НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК *621.391*

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“____” _____ 20*18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *122 Комп'ютерні науки та інформаційні технології*
(код та назва спеціальності)

спеціалізацією *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Методи комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення*

Виконав: студент *VI* курсу групи ІС-63м
(шифр групи)

Романчук Ростислав Олександрович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *проф., д.ф.-м.н., проф. Задірака В. К.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *к.т.н., доц. Жданова О.Г.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018

РЕФЕРАТ

Магістерська дисертація: 103с., 26 рис., 5 табл., 1 додаток, 74 джерел.

Актуальність. Сьогодні нерідко виникає необхідність передати конфіденційне повідомлення невеликого обсягу, при цьому використання складних криптографічних систем по ряду причин важко. Однією з таких причин є неможливість використання надійних продуктів, які, як правило, є комерційними і для рядового користувача комп'ютера недоступні. У сучасному інформаційному суспільстві велика кількість послуг забезпечується за допомогою комп'ютерних мереж та інформаційних технологій. Інформація, що представлена в цифровому вигляді, має бути надійно захищена від багатьох загроз: несанкціонованого доступу та використання, знищення, підробки, витоку, порушення ліцензійних угод, відмови від авторства та ін. Захист інформації є вкрай важливим як в комерційній, так і в державній сферах. Законом України "Про основи національної безпеки України" від 19.06.2003 р. серед загроз національним інтересам і безпеці України в інформаційній сфері зазначені: комп'ютерні тероризм та злочинність; розголошення таємної чи конфіденційної інформації, що є власністю держави або спрямована на забезпечення потреб та національних інтересів суспільства і держави; маніпулювання суспільною свідомістю, зокрема, шляхом поширення недостовірної інформації. Таким чином, питання розроблення ефективних методів захисту цифрової інформації, зокрема методів комп'ютерної стеганографії та стеганоаналізу, актуальні та мають важливе значення для держави й суспільства.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконана на філії кафедри автоматизованих систем обробки інформації та управління в Інституті кібернетики ім. В.М. Глушкова НАН України в рамках науково-дослідної теми «Розробити оптимальні за точністю та швидкодією алгоритми розв'язання задач: інтегрування швидкоосцилюючих функцій, цифрової обробки сигналів та зображень, дистанційного моніторингу об'єктів, інформаційної безпеки» (номер державної реєстрації: 0114U000357).

Мета і завдання дослідження – аналіз стійких до типових операцій обробки методів комп'ютерної стеганографії та методів стеганоаналізу для виявлення найбільш поширених графічних стеганоконтейнерів.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати огляд існуючих стеганографічних алгоритмів;
- здійснити порівняльний аналіз різних стеганографічних алгоритмів;
- запропонувати метод підвищення стеганостійкості;
- визначити ефективність створеного рішення.
- виконати аналіз отриманих результатів.

Об'єкт дослідження – процес захисту інформації, вкрапленої в графічний контейнер.

Предмет дослідження – методи та алгоритми комп'ютерної стеганографії і стеганоаналізу для зображень.

Методи дослідження, застосовані у даній роботі, базуються на стеганографічних алгоритмах.

Наукова новизна одержаних результатів полягає у наступному.

Запропоновано алгоритм комп'ютерної стеганографії для цифрових контейнерів у вигляді зображення, що відрізняється підвищеною ефективністю, який дозволяє здійснювати операції з нанесення тексту на зображення.

Публікації. Матеріали роботи представлено у двох наукових статтях на міжнародних конференціях ISCIENCE 2017 та ISCIENCE 2018, Переяслав-Хмельницький, Україна.

СТЕГАНОГРАФІЯ, ЗАХИСТ ІНФОРМАЦІЇ, СТЕГАНОКОНТЕЙНЕР, ВКРАПЛЕННЯ ІНФОРМАЦІЇ, СТЕГАНОАНАЛІЗ

ABSTRACT

Master dissertation: 103p., 26 ppic., 5 tabl., 1 Add., 74 ref.

Actuality. Today, often occurs a necessity to convey a small-size confidential message, wherein the usage of complex cryptographic systems is difficult for a number of reasons. One of these reasons is the inability to use reliable products, which are usually commercial and are not available for the ordinary computer users. In the modern information society, a large number of services are provided through computer networks and information technologies. The information presented in digital form must be reliably protected against many threats: unauthorized access and usage, destruction, forgery, leakage, violation of license agreements, copyright denial, etc. Information protection is extremely important both in the commercial and public spheres. The Law of Ukraine "About fundamentals of national security of Ukraine" dated 19.06.2003, among the threats to the national interests and security of Ukraine in the information sphere, are: computer terrorism and crime; disclosure of secret or confidential information owned by the state or aimed at ensuring the needs and national interests of society and the state; manipulation of public consciousness, in particular, by disseminating inaccurate information. Thus, the issue of developing effective methods for the protection of digital information, in particular the methods of computer steganography and steganoanalysis, are relevant and of great importance to the state and society.

Connection with academic papers, plans, themes. The work was done at the branch of the department of automated data processing systems and management at the V.M. Glushkov Institute of Cybernetics NAS of Ukraine within the research topic "Development of optimal algorithms for solving problems in accuracy and speed: integration of fast-sensing functions, digital processing of signals and images, remote monitoring of objects, information security" (state registration: 0114U000357).

The goal of the research – analysis of steady-state operations for the processing of computer steganography methods and steganoanalysis methods for the detection of the most common graphic containers.

To achieve the goal the following **tasks** should be performed::

– perform a review of existing steganographic algorithms;

– make a comparative analysis of various steganographic algorithms;

– propose a method of increasing the quiltedness;

– determine the effectiveness of the solution;

– perform the analysis of the results.

The object of the research – the process of protecting information embedded in a graphical container.

Subject of the research – methods and algorithms for computer steganography and steganoanalysis for images.

Research methods, applied in this work, are based on steganographic algorithms.

Scientific novelty of the obtained results is as follows:

The algorithm of computer steganography was proposed for digital containers in the form of an image, which is distinguished by the increased efficiency, allows to make operations on drawing of text on an image.

Publications. The materials of the work were presented in two scientific articles at the international conferences ISCIENCE 2017 and ISCIENCE 2018, Pereyaslav-Khmelnitsky, Ukraine.

STEGANOGRAPHY, INFORMATION PROTECTION, STEGANOCONTAINER, INFORMATION EMBEDDING, STEGANOANALYSIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ

ДІКМ – двовимірна диференціальна імпульсно-кодова модуляція.

ДКП – дискретне косинусне перетворення.

КС – комп'ютерна стеганографія.

НЗБ – найменш значущий біт.

ПЗ – програмне забезпечення.

CPU – англ. Central processing unit – центральний процесор.

RAM – англ. Random Access Memory – оперативна пам’ять комп’ютера.

ВСТУП

Проблема інформаційної безпеки вирішується на протязі всієї історії людства. Ще в давнину виділилося два основні напрямки захисту інформаційних ресурсів: криптографія та стеганографія. Криптографія блокує несанкціонований доступ до даних шляхом їх шифрування. Стеганографія ж іде принципово далі – її мета приховати сам факт існування конфіденційної інформації.

Хоча стеганографія має дуже довгу і багату історію, однак тільки останнім часом у зв'язку з бурхливим розвитком інформаційних технологій, зокрема з появою комп'ютерних мереж, а також через наявність обмежень на використання криптозасобів та надзвичайну актуальність проблеми захисту інтелектуальної власності, стеганографія стає предметом зростаючого інтересу й активних наукових досліджень.

Стеганографічні методи, які приховують інформацію у потоках оцифрованих сигналів та реалізуються на базі комп'ютерної техніки і програмного забезпечення в рамках окремих обчислювальних систем, корпоративних чи глобальних мереж, складають предмет вивчення цифрової стеганографії. Одним із видів стеганографії є стеганофонічні системи – це системи в яких приховується факт передачі таємного повідомлення, а саме повідомлення інкапсулюється у стек мережевих протоколів та передається у реальному масштабі часу. Вперше принципи та визначення комп'ютерної стеганофонії були сформовані польськими спеціалістами з Варшавського університету технологій у 2008 році, які запропонували декілька методів приховування даних у трафіку IP-телефонії.

Структура та принципи роботи систем комп'ютерної стеганофонії аналогічні до стеганографічних систем, тому часто про ці галузі захисту даних порівнюють між собою. Актуальність досліджень у галузі комп'ютерної стеганофонії витікає з обмежень на використання криптографічних засобів та з необхідності розв'язування задач захисту прав власності на інформацію, яка представлена у цифровому вигляді.

На сьогодні в якості інструментів для розвитку цієї галузі широко використовуються методи теорії ймовірностей та математичної статистики, теорії

швидких ортогональних перетворень, теорії апроксимації, теорії кодування, теорії складності, теорії похибок, цифрової обробки сигналів та зображень тощо. Тобто, як бачимо, це вже досить наукоємна дисципліна. Незважаючи на молодість комп'ютерної стеганофонії, основні її поняття та принципи не аналогічні стеганографії. Так в роботах [1-5] наведено базову систему означень та математичні моделі стеганографічних систем. Велика кількість вітчизняних та зарубіжних публікацій присвячена аналізу головної характеристики стегосистеми – її стійкості.

Значний вклад у розвиток стеганофонії та стеганографії внесли такі вчені як: Задірака В.К., Кошкіна Н.В., Олексюк О.С., а також польські вчені Wojciech Mazurczyk, Krzysztof Szczypiorski,Zbigniew Kotulski.

Разом з тим чимало проблем поки що знаходяться на початковій стадії свого вирішення. Наведемо основні з них:

- побудова стійких стеганофонічних систем в рамках моделей пасивного та активного противника;
- отримання оцінок стійкості стеганофонічних систем;
- отримання оцінок складності стеганофонічних алгоритмів.

Вирішення наведених вище проблем забезпечить підвищення стійкості стеганофонічних систем. Досліджено вплив розміру контейнера з прихованими даними на стійкість стегосистеми до виявлення її зловмисником. Запропоновано підхід до вибору оптимальних параметрів стегосистем при заданих мережевих характеристиках, який дає змогу підвищити ефективність та захищеність передачі прихованих даних.

Інформація, що представлена в цифровому вигляді, має бути надійно захищена від багатьох загроз: несанкціонованого доступу та використання, знищення, підробки, витоку, порушення ліцензійних угод, відмови від авторства та ін. Захист інформації є вкрай важливим як в комерційній, так і в державній сферах. Законом України "Про основи національної безпеки України" від 19.06.2003 р. серед загроз національним інтересам і безпеці України в інформаційній сфері зазначені: комп'ютерні тероризм та злочинність; розголошення таємної чи конфіденційної

інформації, що є власністю держави або спрямована на забезпечення потреб та національних інтересів суспільства і держави; маніпулювання суспільною свідомістю, зокрема, шляхом поширення недостовірної інформації. Таким чином, питання розроблення ефективних методів захисту цифрової інформації, зокрема методів комп'ютерної стеганографії та стеганоаналізу, актуальні та мають важливе значення для держави й суспільства. Існує необхідність захисту різних інформаційних систем, зокрема локальних мереж державних та комерційних закладів, від загрози витоку інформації, порушення авторських прав чи особистих таємниць (наприклад, медичних). Не можна виключати й можливість використання здобутків стеганографії антидержавними, терористичними структурами. Тому актуальними і важливими є розроблення та реалізація ефективних методів стеганоаналізу – науки про виявлення стеганографічних приховувань. З огляду на широту практичного застосування та свою гнучкість, найбільші перспективи на сьогодні має універсальний статистичний стеганоаналіз з навчанням та класифікацією. Зважаючи на невпинний розвиток та вдосконалення методів комп'ютерної стеганографії дослідження саме цього напрямку стеганоаналізу є найбільш актуальним.

1 ОГЛЯД СУЧАСНОЇ КОМП'ЮТЕРНОЇ СТЕГАНОГРАФІЇ ТА ЇЇ ПРОБЛЕМАТИКИ

1.1 Генезис становлення поняття «Стеганографія»

Місцем зародження стеганографії багато хто називає Єгипет, хоча першими "стеганографічними повідомленнями" можна назвати і наскальні малюнки древніх людей.

Перша згадка щодо стеганографічних методів в літературі приписується Геродоту, який описав випадок передачі повідомлення Демарто, який зіскоблювавши віск з дощечок, писав лист прямо на дереві, а потім заново покривав дощечки воском.

Інший епізод, який відносять до тих же часів – передача послання з використанням голови раба. Для передачі таємного повідомлення голову раба голили, наносили на шкіру татуювання, і коли волосся відростало, відправляли з посланням[6].

У Китаї листи писали на смужках щілинка. Тому для приховування повідомлень, смужки з текстом листа, згорталися в кульки, покривалися воском і потім ковталися посильними.

Темне середньовіччя породило не тільки інквізицію: посилення стеження привело до розвитку як криптографії, так і стеганографії. Саме в середні століття вперше було застосовано спільне використання шифрів і стеганографічних методів.

У XV столітті чернець Трітеміус (1462-1516), який займався криптографією і стеганографією, описав багато різних методів прихованої передачі повідомлень. Пізніше, в 1499 році, ці записи були об'єднані в книгу "Steganographia", яку в даний час, ті, хто знають латинь можуть прочитати в Інтернет .

XVII - XVIII століття відомі як ера "чорних кабінетів" – спеціальних державних органів по перехопленню, перлюстрації і дешифрування листування. У штат "чорних кабінетів", крім криптографів і дешифрувальників, входили і інші фахівці, в тому числі і хіміки. Наявність фахівців-хіміків було необхідно через активне використання так званих невидимих чорнил. Прикладом може служити цікавий історичний епізод: повсталими дворянами в Бордо був заарештований францисканський монах Берто,

який був агентом кардинала Мазаріні. Повсталі дозволили Берто написати лист знайомого священика в місто Блей. Однак в кінці цього листа релігійного змісту, монах зробив приписку, на яку ніхто не звернув увагу: "Посилаю Вам очну мазь; натріть нею очі і Ви будете краще бачити". Так він зумів переслати не тільки приховане повідомлення, але і вказав спосіб його виявлення. В результаті монах Берто був врятований.

Стеганографічні методи активно використовувалися і в роки громадянської війни між жителями півдня і сіверянами. Так, в 1779 році два агента сіверян Семюель Вудхулл і Роберт Тоунсенд передавали інформацію Джорджу Вашингтону, використовуючи спеціальне чорнило.

Різні симпатичні чорнила використовували і російські революціонери на початку XX століття, що знайшло відображення в радянській літературі: Куканов в своїй повісті "Біля витоків майбутнього" описує застосування молока в якості чорнила для написання таємних повідомлень. Втім, царська охранка теж знала про цей метод (в архіві зберігається документ, в якому описаний спосіб використання симпатичних чорнил і наведено текст перехопленого таємного повідомлення революціонерів).

Особливе місце в історії стеганографії займають фотографічні мікроточки. Так, ті самі мікроточки, які зводили з розуму спецслужби США під час Другої світової війни. Однак мікроточки з'явилися набагато раніше, відразу ж після винаходу Дагером фотографічного процесу, і вперше у військовій справі були використані за часів франко-пруської війни (в 1870 році).

1.2 Стенографія: поняття, особливості, сутність

Стеганографія – це метод організації зв'язку, який власне приховує саму наявність зв'язку [7]. На відміну від криптографії, де ворог точно може визначити чи є передане повідомлення зашифрованим текстом, методи стеганографії дозволяють вбудовувати секретні повідомлення в нешкідливі послання так, щоб неможливо було запідозрити існування вбудованого таємного послання.

Слово "стеганографія" в перекладі з грецької буквально означає "тайнопис" (steganos – секрет, таємниця; graphy – запис). До неї належить величезна безліч секретних засобів зв'язку, таких як невидиме чорнило, мікрофотознімки, умовне розташування знаків, таємні канали та засоби зв'язку на плаваючих частотах і т. д.

Стеганографія займає свою нішу в забезпеченні безпеки: вона не замінює, а доповнює криптографію. Приховування повідомлення методами стеганографії значно знижує ймовірність виявлення самого факту передачі повідомлення. А якщо це повідомлення до того ж зашифровано, то воно має ще один, додатковий, рівень захисту [8-10].

В даний час у зв'язку з бурхливим розвитком обчислювальної техніки і нових каналів передачі інформації з'явилися нові стеганографічні методи, в основі яких лежать особливості подання інформації в комп'ютерних файлах, обчислювальних мережах і т. п. Це дає нам можливість говорити про становлення нового напряму – комп'ютерної стеганографії.

Незважаючи на те що стеганографія як спосіб приховування секретних даних відома вже протягом тисячоліть, комп'ютерна стеганографія – молодий напрямок.

Як і будь-який новий напрямок, комп'ютерна стеганографія, незважаючи на велику кількість відкритих публікацій і щорічні конференції, довгий час не мала єдиної термінології.

До недавнього часу для опису моделі стеганографічної системи використовувалася запропонована 1983 році Сіммонс так звана "проблема ув'язнених". Вона полягає в тому, що два індивідуума (Аліса і Боб) хочуть обмінюватися секретними повідомленнями без втручання охоронця (Віллі), контролюючого комунікаційний канал. При цьому є ряд припущень, які роблять цю проблему більш-менш розв'язуваною. Перше припущення полегшує вирішення проблеми і полягає в тому, що учасники інформаційного обміну можуть розділяти секретне повідомлення (наприклад, використовуючи кодову клавішу) перед укладанням. Інше допущення, навпаки, ускладнює вирішення проблеми, так як охоронець має право не тільки читати повідомлення, але і модифікувати (змінювати) їх.

Пізніше, на конференції Information Hiding: First Information Workshop в 1996 році було запропоновано використовувати єдину термінологію і обговорені основні терміни.

К. Шеннон дав нам загальну теорію тайнопису, яка є базисом стеганографії як науки. У сучасній комп'ютерній стеганографії існує два основних типи файлів: повідомлення-файл, який призначений для приховування, і контейнер-файл, який може бути використаний для приховування в ньому повідомлення. При цьому контейнери бувають двох типів. Контейнер-оригінал (або "порожній" контейнер) - це контейнер, який не містить прихованої інформації. Контейнер-результат (або "заповнений" контейнер) - це контейнер, який містить приховану інформацію. Під ключем розуміється секретний елемент, який визначає порядок занесення повідомлення в контейнер.

Основними положеннями сучасної комп'ютерної стеганографії є наступні [11-15]:

- методи приховування повинні забезпечувати автентичність і цілісність файлу.
- передбачається, що противнику повністю відомі можливі стеганографічні методи.
- безпека методів ґрунтується на збереженні стеганографічних перетворень основних властивостей відкрито переданого файлу при внесенні до нього секретного повідомлення і деякої невідомої противнику інформації - ключа.
- навіть якщо факт приховування повідомлення став відомий противнику через спільника, витяг самого секретного повідомлення являє складну обчислювальну задачу.

У зв'язку зі зростанням ролі глобальних комп'ютерних мереж стає все більш важливим значення стеганографії. Аналіз інформаційних джерел комп'ютерної мережі Internet дозволяє вставити висновок, що в даний час стеганографічні системи активно використовуються для вирішення наступних основних завдань:

- захист конфіденційної інформації від несанкціонованого доступу;

– подолання систем моніторингу та управління мережевими ресурсами;

– камуфлювання програмного забезпечення;

– захист авторського права на деякі види інтелектуальної власності.

Захист конфіденційної інформації від несанкціонованого доступу.

Це область використання КС є найбільш ефективною при вирішенні проблеми захисту конфіденційної інформації. Так, наприклад, тільки одна секунда оцифрованого звуку з частотою дискретизації 44100 Гц і рівнем відліку 8 біт в стерео режимі дозволяє приховати за рахунок заміни найменш значущих молодших розрядів на приховуване повідомлення близько 10 Кбайт інформації. При цьому, зміна значень відліків складає менше 1%. Така зміна практично не виявляється при прослуховуванні файлу більшістю людей [16].

Подолання систем моніторингу та управління мережевими ресурсами.

Стеганографічні методи, спрямовані на протидію системам моніторингу і управління мережевими ресурсами промислового шпигунства, дозволяють протистояти спробам контролю над інформаційним простором при проходженні інформації через сервери керування локальних і глобальних обчислювальних мереж.

Камуфлювання програмного забезпечення (ПЗ).

Іншим важливим завданням стеганографії є камуфлювання ПЗ. У тих випадках, коли використання ПЗ незареєстрованими користувачами є небажаним, воно може бути закамуфльовано під стандартні універсальні програмні продукти (наприклад, текстові редактори) або приховано в файлах мультимедіа (наприклад, в звуковому супроводі комп'ютерних ігор).

Захист авторських прав.

Ще однією областю використання стеганографії є захист авторського права від піратства. На комп'ютерні графічні зображення наноситься спеціальна мітка, яка залишається невидимою для очей, але розпізнається спеціальним ПЗ. Таке програмне забезпечення вже використовується в комп'ютерних версіях деяких журналів. Даний напрямок стеганографії призначений не тільки для обробки зображень, але і для файлів з аудіо-та відео і покликане забезпечити захист інтелектуальної власності [17].

Стеганографічна система або стегосистеми – сукупність засобів і методів, які використовуються для формування прихованого каналу передачі інформації.

При побудові стегосистеми повинні враховуватися наступні положення:

– противник має повне уявлення щодо стеганографічної системі і деталях її реалізації. Єдиною інформацією, яка залишається невідомою потенційному супротивникові, є ключ, за допомогою якого тільки його власник може встановити факт присутності і зміст прихованого повідомлення;
– якщо противник якимось чином дізнається про факт існування прихованого повідомлення, це не повинно дозволити йому отримати подібні повідомлення в інших даних до тих пір, поки ключ зберігається в таємниці;
– потенційний противник повинен бути позбавлений будь-яких технічних та інших переваг в розпізнаванні або розкритті змісту таємних повідомлень.

Узагальнена модель стегосистеми представлена на рисунку 1.1.

Рисунок 1.1 – Узагальнена модель стегосистеми

В якості даних може використовуватися будь-яка інформація: текст, повідомлення, зображення і т. п.

У загальному ж випадку доцільно використовувати слово "повідомлення", так як повідомленням може бути як текст або зображення, так і, наприклад, звукові дані. Далі для позначення інформації, що приховується, будемо використовувати саме термін повідомлення.

Контейнер – будь-яка інформація, призначена для приховування таємних повідомлень.

Порожній контейнер – контейнер без вбудованого повідомлення; заповнений контейнер або стеганоконтейнер, що містить вбудовану інформацію.

Вбудоване (приховане) повідомлення – повідомлення, що вбудовується в контейнер.

Стеганографічний канал або просто стеганоканал – канал передачі стеганоконтейнера.

Стеганоключ або просто ключ – секретний ключ, необхідний для приховування інформації. Залежно від кількості рівнів захисту (наприклад, вбудовування попередньо зашифрованого повідомлення) у стеганосистемі може бути один або кілька стеганоключів [18].

За аналогією з криптографією, за типом стегоключа стегосистеми можна поділити на два типи:

– з секретним ключем;
– з відкритим ключем.

У стеганосистем з секретним ключем використовується один ключ, який повинен бути визначений або до початку обміну секретними повідомленнями, або переданий по захищеному каналу.

У стеганосистеми з відкритим ключем для вбудовування і вилучення повідомлення використовуються різні ключі, які розрізняються таким чином, що за допомогою обчислень неможливо вивести один ключ з іншого. Тому один ключ (відкритий) може передаватися вільно по незахищеному каналу зв'язку. Крім того, дана схема добре працює і при взаємній недовірі відправника і одержувача.

Будь-яка стеганосистема повинна відповідати наступним вимогам [19-21]:

– властивості контейнера повинні бути модифіковані, щоб зміну неможливо було виявити при візуальному контролі. Ця вимога визначає якість приховування впроваджуваного повідомлення: для забезпечення безперешкодного проходження стеганоповідомлення по каналу зв'язку воно жодним чином не повинно привернути увагу атакуючого.

– стеганоповідомлення має бути стійким до спотворень, в тому числі і зловмисним. В процесі передачі зображення (звук або інший контейнер) може зазнавати різні трансформації: зменшуватися або збільшуватися, перетворюватися в інший формат і т. д. Крім того, воно може бути стисло, в тому числі і з використанням алгоритмів стиснення з втратою даних.
– для збереження цілісності вбудованого повідомлення необхідно використання коду з виправленням помилки.
– для підвищення надійності повідомлення, що вбудовується повинно бути продубльовано.

В даний час можна виділити три тісно пов'язаних між собою і таких, що мають одне коріння напрями докладання стеганографії: приховування даних (повідомлень), цифрові водяні знаки і заголовки.

Приховування впроваджуваних даних, які в більшості випадків мають великий обсяг, пред'являє серйозні вимоги до контейнера: розмір контейнера в кілька разів повинен перевищувати розмір вбудованих даних.

Цифрові водяні знаки використовуються для захисту авторських або майнових прав на цифрові зображення, фотографії або інші оцифровані твори мистецтва. Основними вимогами, які пред'являються до таких вбудованих даних, є надійність і стійкість до спотворень.

Цифрові водяні знаки мають невеликий обсяг, проте, з урахуванням зазначених вище вимог, для їх вбудовування використовуються більш складні методи, ніж для вбудовування просто повідомлень або заголовків. В даному випадку стеганографічні методи використовуються не тільки для впровадження ідентифікуючого заголовка, але і інших індивідуальних ознак файлу [22].

Впроваджувані заголовки мають невеликий обсяг, а вимоги до них мінімальні: заголовки повинні вносити незначні спотворення і бути стійкі до основних геометричних перетворень.

Кожен з перерахованих вище додатків вимагає певного співвідношення між стійкістю вбудованого повідомлення до зовнішніх впливів (в тому числі і стеганоаналізу) і розміром самого вбудованого повідомлення.

Для більшості сучасних методів, використовуваних для приховування повідомлення в цифрових контейнерах, має місце наступна залежність надійності системи від обсягу вбудованих даних, як показано на рисунку 1.2.

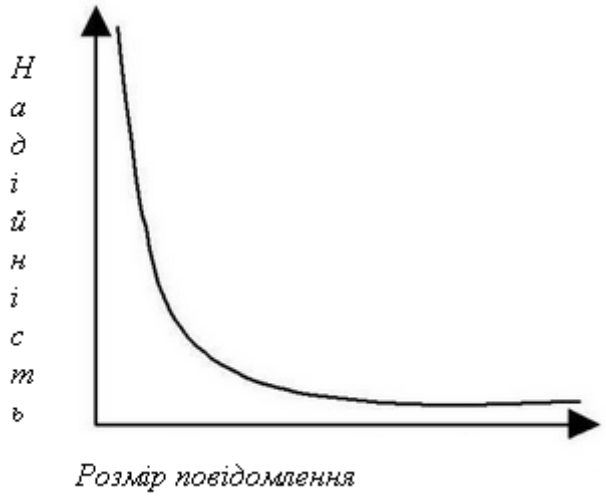

Рисунок 1.2 – Залежність надійності системи від обсягу вбудованих даних

Дана залежність показує, що при збільшенні обсягу вбудованих даних знижується надійність системи (при незмінності розміру контейнера). Таким чином, використовуваний в стеганосистемі контейнер накладає обмеження на розмір вбудованих даних.

1.3 Контейнери

Істотний вплив на надійність стеганосистеми і можливість виявлення факту передачі прихованого повідомлення надає вибір контейнера. Наприклад, досвідчене око цензора з художньою освітою легко виявить зміну колірної гами при впровадженні повідомлення в репродукцію «Мадонни» Рафаеля або "Чорного квадрата" Малевича.

За протяжністю контейнери можна поділити на два типи: безперервні (потокові) і обмеженої (фіксованої) довжини. Особливістю потокового контейнера є те, що неможливо визначити його початок або кінець [23]. Більш того, немає

можливості дізнатися заздалегідь, якими будуть наступні шумові біти, що призводить до необхідності включати приховуючи повідомлення біти в потік в реальному масштабі часу, а самі приховуючи біти вибираються за допомогою спеціального генератора, що задає відстань між послідовними бітами в потоці.

У безперервному потоці даних найбільша трудність для одержувача – визначити, коли починається приховане повідомлення. При наявності в потоковому контейнері сигналів синхронізації або кордонів пакета, приховане повідомлення починається відразу після одного з них. У свою чергу, для відправника можливі проблеми, якщо він не впевнений в тому, що потік контейнера буде достатньо довгим для розміщення цілого таємного повідомлення.

При використанні контейнерів фіксованої довжини відправник заздалегідь знає розмір файлу і може вибрати приховуючи біти у псевдовипадковій послідовності. З іншого боку, контейнери фіксованої довжини, як це вже зазначалося вище, мають обмежений обсяг і іноді повідомлення, що вбудовується може не помістится в файл-контейнері.

Інший недолік полягає в тому, що відстані між приховуючими бітами рівномірно розподілені між найбільш короткими і найбільш довгими заданими відстанями, в той час як справжній випадковий шум буде мати експоненціальний розподіл довжин інтервалу. Звичайно, можна породити псевдовипадкові експоненціально розподілені числа, але цей шлях зазвичай занадто трудомісткий. Однак на практиці найчастіше використовуються саме контейнери фіксованої довжини, як найбільш поширені і доступні. Можливі наступні варіанти контейнерів [24]:

– контейнер, що генерується самою стеганосистемою. Прикладом може служити програма MandelSteg, в якій в якості контейнера для вбудовування повідомлення генерується фрактал Мандельброта. Такий підхід можна назвати конструюючою стеганографією.
– контейнер вибирається з деякої безлічі контейнерів. В цьому випадку генерується велика кількість альтернативних контейнерів, щоб потім вибрати

найбільш підходящий для приховування повідомлення. Такий підхід можна назвати селектуючою стеганографією. В даному випадку при виборі оптимального контейнера з безлічі згенерованих найважливішою вимогою є природність контейнера. Єдиною ж проблемою залишається те, що навіть оптимально організованого контейнеру дозволяє заховати незначну кількість даних при дуже великому обсязі самого контейнера.

– контейнер надходить ззовні. В даному випадку відсутня можливість вибору контейнера і для приховування повідомлення береться перший-ліпший контейнер, що не завжди підходить для вбудованого повідомлення. Назвемо це безальтернативною стеганографією.

1.4 Проблематика та постановка завдання дослідження

В даний час, поряд з широким використанням цифрових форматів мультимедіа та існуючими проблемами управління цифровими ресурсами, стають все більш актуальними дослідження в області стеганографії [1 - 6]. Рішення завдання приховування інформації також є важливою проблематикою в умовах розвиненої інфраструктури мережевого спілкування користувачів глобальних комп'ютерних мережах, з розвитком яких стало можливим швидко і економічно вигідно передавати електронні документи в різні куточки планети. При цьому значні обсяги переданих матеріалів часто супроводжуються незаконним копіюванням та розповсюдженням. Як наслідок, це змушує шукати способи приховування авторської інформації в різних текстових, графічних, аудіо, відео, та інших типах файлів.

На сьогоднішній день існує досить багато програмних продуктів, які застосовуються для цілей стеганографії і реалізують методи впровадження конфіденційних даних в різні типи файлів. Схему типової стеганосистеми показано у додатку А(Плакат 1).

Класична задача стеганографії полягає в організації передачі секретного повідомлення таким чином, щоб як зміст повідомлення, так і сам факт його передачі були приховані від усіх, крім зацікавлених осіб. Для вирішення такого завдання

використовується деяке повідомлення, зване контейнером (стеганоконтейнером), в який вбудовується необхідне для передачі секретне повідомлення. При цьому розробники стеганографічних методів повинні організувати прозорість переданих конфіденційних даних: зміна певного числа інформаційних біт в контейнері не повинна привести до особливих втрат його якості (повинні бути відсутніми артефакти візуалізації вбудовування). У якості контейнерів найбільш часто виступають файли, що містять цифрові фотографії, текст, музику, відео. Так, наприклад, при використанні в якості контейнера графічних файлів для сторонніх спостерігачів процес передачі повідомлень буде сприйматися як звичайний обмін цифровими графічними файлами. Слід при цьому пам'ятати про важливість дотримання однієї умови: ніхто не повинен мати доступ одночасно до вихідного файлу, обраного в якості контейнера, і до файлу, який містить приховане повідомлення, тому що в такому випадку просте порівняння файлів відразу ж виявить наявність повідомлення. Як було зазначено вище, в комп'ютерній стеганографії у якості контейнеру може виступати практично будь-який файловий формат, проте найбільш поширеним типом носія є файли зображення формату BMP. Це пояснюється тим, що для цілей стеганографії найкращими є файли форматів, в яких використовуються методи стиснення без втрат (такі види стиснення типові для зображень формату BMP, TIFF, PNG, TGA, і ін.). Також позитивною стороною на користь вибору формату BMP виступає висока якість зображення і простота формату.

Метою даної магістерської роботи виступає дослідження методів комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення та практична реалізація вдосконаленого методу комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення.

Для досягнення поставленої мети у роботі пропонуються до вирішення наступні завдання:

– всебічне дослідження методів комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення;

- формалізація головних недоліків комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення;
- вдосконалення методу комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення;
- практична реалізація вдосконаленого методу комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення.

Висновки до розділу

У рамках першого розділу даної магістерської роботи висвітлено теоретичні питання реалізації стеганографічних методів. Окреслено генезис становлення поняття стеганографія, здійснено дослідження основних складових та компонентів реалізації методів комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення. Також здійснено формулювання загальної мети дослідження та структуризація завдань досягнення поставленої мети.

2 МОДЕЛІ ТА МЕТОДИ КОМП’ЮТЕРНОЇ СТЕГАНОГРАФІЇ ДЛЯ ЦИФРОВИХ КОНТЕЙНЕРІВ У ВИГЛЯДІ ЗОБРАЖЕННЯ

2.1 Загальний огляд стеганографічних методів

В даний час методи комп'ютерної стеганографії розвиваються за двома основними напрямками [25]:

– методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

– методи, засновані на надмірності аудіо та візуальної інформації.

Порівняльні характеристики існуючих стеганографічних методів наведені в табл. 2.1.

Таблиця 2.1 – Порівняльні характеристики стеганографічних методів

Стеганографічні методи	**Коротка характеристика методів**	**Недоліки**	**Переваги**
1. Методи використання спеціальних властивостей комп'ютерних форматів даних			
1.1. Методи використання зарезервованих для розширення полів комп'ютерних форматів даних	Поля розширення є в багатьох мультимедійних форматах, вони заповнюються нульовий інформацією і не враховуються програмою	Низька ступінь скритності, передача невеликих обмежених обсягів інформації	Простота використання
1.2. Методи спецформатування текстових файлів:			
1.2.1. Методи використання відомого зміщення слів, речень, абзаців	Методи засновані на зміні положення рядків і розстановки слів у реченні, що забезпечується вставкою додаткових пробілів між словами	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Простота використання. Є опубліковане програмне забезпечення реалізації даного методу
1.2.2. Методи вибору певних позицій букв (нульовий шифр)	Без цензури - окремий випадок цього методу (наприклад, початкові літери кожного рядка утворюють повідомлення)		
1.2.3. Методи використання спеціальних властивостей полів форматів, які не відображаються на екрані	Методи засновані на використанні спеціальних "невидимих", прихованих полів для організації виносок і посилань (наприклад, використання чорного шрифту на чорному тлі)		

Продовження таблиці 2.1

1.3. Методи приховування в невикористовуваних місцях гнучких дисків	Інформація записується в зазвичай невикористовуваних місцях ГМД (наприклад, в нульовий доріжці)	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Простота використання. Є опубліковане програмне забезпечення реалізації даного методу
1.4. Методи використання імітуючих функцій (mimic-function)	Метод заснований на генерації текстів і є узагальненням акровірша. Для таємного повідомлення генерується осмислений текст, що приховує саме повідомлення	1. Слабка продуктивність методу, передача невеликих обсягів інформації 2. Низька ступінь скритності	Результуючий текст не є підозрілим для систем моніторингу мережі
1.5. Методи видалення ідентифікуючий файл заголовка	Приховуване повідомлення шифрується і у результаті видаляється ідентифікуючий заголовок, залишаючи тільки шифровані дані. Одержувач заздалегідь знає про передачу повідомлення і має недостатній заголовок	Проблема приховування вирішується тільки частково. Необхідно заздалегідь передати частину інформації одержувачу	Простота реалізації. Багато засобів (White Noise Storm, S-Tools), забезпечують реалізацію цього методу з PGP шифроалгоритмом
2. Методи використання надмірності аудіо та візуальної інформації			
2.1. Методи використання надмірності цифрових фотографії, цифрового звуку і цифрового відео	Молодші розряди цифрових відліків містять дуже мало корисної інформації. Їх заповнення додатковою інформацією практично не впливає на якість сприйняття, що і дає можливість приховування конфіденційної інформації	За рахунок введення додаткової інформації спотворюються статистичні характеристики цифрових потоків. Для зниження компрометуючих ознак потрібна корекція статистичних характеристик	Можливість прихованої передачі великого обсягу інформації. Можливість захисту авторського права, прихованого зображення товарної марки, реєстраційних номерів і т.п.

Як видно з табл. 2.1, перший напрямок засновано на використанні спеціальних властивостей комп'ютерних форматів представлення даних, а не на надмірності самих даних. Спеціальні властивості форматів вибираються з урахуванням захисту приховуваного повідомлення від безпосереднього прослуховування, перегляду або прочитання. На підставі аналізу матеріалів табл. 2.1 можна зробити висновок, що

основним напрямком комп'ютерної стеганографії є використання надмірності аудіо та візуальної інформації. Цифрові фотографії, цифрова музика, цифрове відео – представляються матрицями чисел, які кодують інтенсивність в дискретні моменти в просторі і / або в часі. Цифрова фотографія – це матриця чисел, що представляють інтенсивність світла в певний момент часу. Цифровий звук – це матриця чисел, що представляє інтенсивність звукового сигналу в моменти часу, що йдуть послідовно. Всі ці цифри не точні, тому що не точні пристрої оцифровки аналогових сигналів, є шуми квантування. Молодші розряди цифрових відліків містять дуже мало корисної інформації про поточні параметри звуку і візуального образу. Їх заповнення відчутно не впливає на якість сприйняття, що і дає можливість для приховування додаткової інформації [26].

Графічні кольорові файли зі схемою змішування RGB кодують кожну точку малюнка трьома байтами. Кожна така точка складається з адитивних складових: червоного, зеленого, синього. Зміна кожного з трьох найменш значущих біт приводить до зміни менш 1% інтенсивності даної точки. Це дозволяє приховувати в стандартному графічному зображенні об'ємом 800 Кбайт близько 100 Кбайт інформації, що не видно при перегляді зображення.

2.2 Методи приховування інформації в графічних зображеннях

Всі методи, призначені для приховування даних, можна розділити за принципам, які лежать в їх основі, на форматні та неформатні.

Форматні методи приховування (форматні стеганографічні системи) – це такі методи (системи), які ґрунтуються на особливостях формату зберігання графічних даних. Розробка таких методів зводиться до аналізу формату з метою пошуку службових полів формату, зміна яких в конкретних умовах не позначиться на роботі з графічним зображенням. Наприклад, для приховування можна використовувати службові поля формату, які присутні в графічних файлах, але не використовуються в даний час [27,28]. Однак всі форматні методи мають загальний недолік – для них можлива побудова повністю автоматичного алгоритму, спрямованого на виявлення

факту приховування (з урахуванням принципу загальновідомості стеганографічної системи). Тому їх стійкість до атак пасивних супротивників вкрай низька.

Неформатні методи – це методи, які використовують безпосередньо самі дані, якими зображення представлено в цьому форматі. Застосування неформатних методів неминуче призводить до появи спотворень, що вносяться стеганографічною системою, однак при цьому вони є більш стійкими до атак як пасивних, так і активних противників.

2.2.1 Неформатні методи приховування в графічних зображеннях

Неформатні методи приховування в JPEG.

В результаті детального аналізу алгоритму стиснення з втратами JPEG , режимів його роботи і проміжних етапів (таких як перетворення зображення в оптимальний колірний простір, субдискретизація, дискретне косинусне перетворення, квантування і кодування) були розроблені методи, що дозволяють виробляти неформатне приховування даних в файли, формати яких побудовані відповідно до специфікації JPEG.

Метод приховування у вихідних даних зображення.

Стандарт JPEG дозволяє виробляти стиснення зображень без втрат (режим Lossless JPEG), цей режим істотно відрізняється від режиму з втратами, заснованого на базі квантування коефіцієнтів дискретного косинусного перетворення (ДКП). Lossless JPEG являє собою кодування з пророкуванням (використовує схему двовимірної диференціальної імпульсно-кодової модуляції – ДІКМ), коли значення кожного пікселя об'єднується зі значеннями сусідніх з ним для формування величини прогнозуючого параметра. Потім отриманий результат віднімається з вихідного значення. Сформовані після обробки так само всіх точок зображення результуючі величини стискаються за допомогою арифметичного кодування або кодування за методом Хаффмана [29].

Тому в разі використання Lossless JPEG можна говорити про приховування інформації безпосередньо в даних самого зображення. При цьому приховування може

здійснюватися за допомогою основного методу комп'ютерної стеганографії – методу приховування в молодших бітах і його модифікацій.

Однак Lossless JPEG застосовується вкрай рідко на практиці, а використовувати формат JPEG в режимі стиснення з втратами при приховуванні інформації даним методом можна, так як вона буде втрачена в силу особливостей використовуваного алгоритму, такого як субдискретизація, ДКП, квантування.

Метод приховування з використанням таблиць квантування.

Даний метод є одним з найчастіше використовуваних на сьогодні методів приховування даних файлів JPEG. Ідея полягає у використанні для приховання молодших бітів чисел, що представляють коефіцієнти квантування. Гідність методу полягає в тому, що він не порушує типову структуру потоку JPEG і, отже, є повністю неформатним. До недоліків можна віднести те, що зазвичай файли JPEG містять одну або дві таблиці квантування (розмір однієї таблиці квантування дорівнює 64 байтам), тому обсяг приховуваних даних невеликий (приховування у всіх молодших бітах однієї таблиці квантування дозволяє приховати всього лише 8 байт). Крім цього зміна молодших біт коефіцієнтів квантування вносить зміни в статистичні характеристики блоків, що стискаються, тим самим негативно впливаючи на ефективність подальшого кодування, і, як наслідок, веде до збільшення розмірів файлу.

Метод використання неправдивих таблиць квантування.

Метод є подальшим розвитком попереднього методу. Він полягає у створенні додаткових помилкових таблиць квантування. Це дозволяє у кілька разів збільшити обсяг приховуваних даних в порівнянні з попереднім методом. У стандарті JPEG врахована можливість використання декількох таблиць квантування, тобто це не порушує внутрішню організацію формату. Однак крім того, що для даного методу зберігаються зазначені вище недоліки, він стає частково форматним, так як використовується особливість формату, яка є допустимою, але не типовою. Взагалі кажучи, на практиці застосовуються два різновиди методу використання неправдивих таблиць квантування [30].

Перший різновид додає таблиці так, щоб збільшити ефективність стиснення і зменшити втрати при стисненні, як це і передбачалося в специфікації алгоритму

JPEG. Однак в такому випадку для більшості зображень число помилкових таблиць невелике.

Другий різновид полягає в додаванні помилкових таблиць квантування з певним (не завжди фіксованим) періодом, при цьому використовуються, як правило, одні і ті ж таблиці, відмінності яких складаються лише в тих молодших бітах, де сховано повідомлення [31]. Природно, цей метод є форматним і не має стійкості до атак пасивного противника, спрямованих на визначення факту наявності прихованого повідомлення.

Метод приховування в спектрі зображення після квантування.

Метод заснований на використанні частот блоків зображення після їх квантування, але перед етапом кодування. При цьому приховування може здійснюватися за допомогою класичних методів комп'ютерної стеганографії. Даний метод дозволяє приховувати набагато більше число біт, ніж наведені вище методи, і не є форматним, тому його стійкість до атаки пасивного противника може значно (в залежності від реалізації) перевищувати рівень стійкості наведених раніше методів. При використанні даного методу обсяг приховуваних даних пропорційний обсягу стисненого зображення, при цьому збільшення обсягу впроваджуваної інформації може призводити до змін вихідного зображення і зниження ефективності подальшого етапу кодування. Однак можливість варіювати якість стисненого зображення в широкому діапазоні не дозволяє легко встановити, виникаючі в результаті стиснення похибки є наслідком приховування даних або використання великих коефіцієнтів квантування.

Суть методу полягає в наступному. Нехай m – біти приховуваного повідомлення, $B_{i,\ j}$ - значення ненульових елементів блоків квантованного спектра немодифікованого зображення, впорядковані згідно з порядком їх кодування в алгоритмі JPEG, де i – номер біта елемента, j – номер елемента, $B_{i,\ j}'$ – відповідні блоки модифікованого зображення [32].

Формується двійкова послідовність k_j, значення бітів якої відповідають блокам $B_{i,\ j}$, при цьому $k_j = 1$, коли в молодший біт j -го блоку ховається черговий біт

повідомлення, і $k_j = 0$ в іншому випадку. Тоді пряме стеганографічне перетворення F: M x B x K→B для даного методу має наступний вигляд:

$$B'_{i,j} = \begin{cases} B_{i,j}, & \forall_i, k_j = 0 \\ m_l, i = 0, k_j = 1 \end{cases}$$

де $l = \sum_{p=1}^{j} k_p;\ j = 1,2,3, \dots n$, а відповідне йому зворотне стеганографічне перетворення $F^{-1}: BxK \to M$ має вигляд $m_j = B_{0,i}^{l}$

де l таке, що $l = \sum_{p=1}^{j} k_p = j, j = 1,2,3, \dots n$

Методи приховування в графічних зображеннях з палітрою кольорів.

Використання палітри (ще говорять - відображення кольорів) в графічних форматах пов'язано зі спробою зменшити розмір, що зберігається. Взагалі кажучи, палітра вперше була застосована в графічних адаптерах для спрощення їх устрою та забезпечення більшого дозволу при меншому обсязі оперативної пам'яті графічного адаптера [33]. Слідом за цим з'явилися формати зберігання растрових графічних зображень, засновані на використанні палітри, деякі з яких активно використовуються і в наші дні. Яскравим прикладом такого формату може служити GIF, який набув широкого поширення в мережі Інтернет і є невід'ємною частиною дизайну сучасних веб-сторінок і Інтернет-реклами [34]. Кількість переданих по мережі файлів у форматі GIF більш ніж в два рази перевищує кількість переданих сторінок і листів (на зміну застарілого формату був розроблений формат PNG, який також дозволяє використовувати палітру кольорів, проте він ще не набув великого поширення).

Кожна точка звичайного реєстрового графічного зображення задає інтенсивність колірних складових в будь-якому фіксованому колірному просторі (RGB, CMY, CMYK і т.д.). Якщо ж формат зберігання зображень використовує палітру, то точки зображення можуть приймати лише один колір з наявних в палітрі [35]. Палітра кольорів – це набір з елементів, кожен з яких задає (як і точка звичайного зображення) інтенсивність колірних складових в будь-якому фіксованому колірному

просторі (зазвичай RGB), при цьому кожна точка зображення містить лише номер кольору з палітри, а не інформацію про її кольор в колірному просторі.

Нижче представлений приклад 8-бітного RGB-зображення розміром 4x4 точки, що складається з точок зеленого і синього кольору, які чергуються в шаховому порядку. Зображення записано у вигляді матриці з (R, G, B) – елементами :

(0,255,0) (0,0,255) (0,255,0) (0,0,255)
(0,0,255) (0,255,0) (0,0,255) (0,255,0)
(0,255,0) (0,0,255) (0,255,0) (0,0,255)
(0,0,255) (0,255,0) (0,0,255) (0,255,0)

Для зберігання такої матриці необхідно 384 біт пам'яті.

Якщо використовувати зображення з палітрою, то для даного зображення потрібна палітра, що складається з двох кольорів:

$$0 \rightarrow (0,255,0); 1 \rightarrow (0,0,255)$$

Тоді в цій палітрі зображення набуде вигляду

0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

Таким чином, в пам'яті необхідно 48 біт для зберігання інформації про використовувану палітру і 16 біт для зберігання самого зображення.

Метод приховування з використанням молодших біт даних зображення.

З наведеного вище прикладу видно, що використовувати метод приховування в молодших бітах для зображень з палітрою без додаткового доопрацювання не можна, так як елементи палітри, номер яких відрізняється лише молодшим бітом, можуть мати абсолютно різні кольори, і тому зміни молодшого біта можуть привести до помітних змін самого зображення.

Найпростіший спосіб, що дозволяє подолати ці труднощі, полягає в тому, що перед приховуванням повідомлення в молодших бітах зображення здійснюється аналіз палітри зображення. Серед всіх пар (2i, 2i + 1) елементів палітри здійснюється пошук пар, різниця між колірними інтенсивностями яких не перевищує заданої порогової величини d. Приховування здійснюється в молодші біти тільки тих точок

зображення, які посилаються на відібрані елементи палітри. Так як при приховуванні палітра не змінюється, то перш ніж вийняти проводиться її аналіз аналогічним чином.

Однак кількість пар елементів палітри, придатних для приховування таким способом, як правило, не велика. Даний метод можна поліпшити, додавши перед аналізом палітри її сортування зі збереженням старих номерів елементів по зростанню ваги, рівної, наприклад, (65536R + 256G + B). Пари, придатні для приховування, відбираються за допомогою аналізу відсортованої палітри, але приховування в даному випадку проводиться дещо іншим способом. Тепер для приховування біта повідомлення необхідно змінити не молодший біт точки зображення, а все її значення на нове, яке виходить шляхом зміни з молодшого біта номера відсортованої палітри в тому випадку, якщо він придатний для приховування. Інакше кажучи, з кожним елементом палітри тепер пов'язані два числа i, j_i , де i - номер i -го елементу палітри, отриманий в результаті її сортування. Приховування полягає в тому, що послідовно проглядаються всі крапки зображення, за значенням точки k визначається відповідний номер j_k. Якщо номер j_k придатний для приховування, то його молодший біт замінюється на черговий біт повідомлення. Потім з j_k визначається пов'язаний з ним вихідний номер k, який і присвоюється поточній точці [36].

Метод приховування з використанням молодших біт елементів палітри.

У всіх форматах, які використовують палітру кольорів, сама палітра повинна зберігатися разом із зображенням в його файлі, а, отже, для приховування можна використовувати метод приховування в молодших бітах елементів палітри (так як формат зберігання елемента палітри аналогічний формату зберігання точки звичайного зображення без палітри) . Однак розмір палітри не перевищує 256 елементів, в кожен з яких можна приховати не більше 3 біт. Тобто даним методом можна приховати повідомлення розміром не більше 768 біт (що значно менше розміру самого зображення). Крім того, в результаті приховування в палітрі можуть з'явитися елементи, які кодують однакові кольори. Наявність таких «однакових» елементів в палітрі зображення може використовуватися в якості критерію для визначення факту наявності повідомлення, прихованого в молодших бітах палітри.

Метод приховування, заснований на наявності однакових елементів палітри.

Ясно, що в разі, коли палітра містить два або більше однакових елементів, не важливо, який з них буде присвоєно точці зображення, так як при перегляді вона буде виглядати однаково. Цю особливість можна використовувати для приховування, не вносячи жодних викривлень в саме зображення. Відзначимо, що з точки зору кодування графічних зображень використання однакових елементів палітри не тільки позбавлене сенсу, але і небажано, оскільки може призводити до збільшення розміру зображення [37]. З точки зору стеганографії метод, заснований на використанні однакових кольорів палітри, є форматним, так як базується на використанні особливості, якою володіють всі графічні формати, які підтримують палітру кольорів.

У загальному випадку цей метод зводиться до пошуку кількох елементів палітри з найбільшою частотою появи в графічному зображенні. У палітру додаються їх «двійники», після чого послідовно проглядаються всі крапки зображення. Якщо точка посилається на елемент, який має «двійника», то вона використовується для приховування чергового біта повідомлення (наприклад, якщо біт повідомлення дорівнює 1, то значення точки замінюється на «двійника»).

Нижче розглянуто приклад використання даного методу. Нехай повідомлення m = 10010110, палітра складається з двох кольорів:

$$0 \rightarrow (0,255,0); 1 \rightarrow (0,0,255)$$

і зображення має вигляд

0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

Після додавання в палітру елемента $2 \rightarrow (0,255,0)$ в зображенні можна приховати повідомлення m (Виділено жирним шрифтом):

2 1 **0** 1
1 **0** 1 **2**
0 1 **2** 1
1 **2** 1 **0**

Цей метод можна легко продовжити на випадок додавання декількох однакових кольорів, проте при цьому зменшиться його стійкість.

Метод приховування шляхом перестановки елементів палітри.

Ідея даного методу полягає у використанні порядку елементів палітри зображення для приховування інформації. Будемо припускати, що палітра довільного фіксованого зображення складається з n різних елементів, тобто серед них немає жодної пари однакових. З комбінаторики відомо, що кількість перестановок n різних елементів одно n!. Легко зрозуміти, що якщо використовувати перестановки для приховування довічного повідомлення, то його максимальна довжина складе близько log_2 (n!) біт. У загальному випадку метод приховування шляхом перестановки елементів палітри полягає в тому, що задається відображення, яке при фіксованому ключі взаємно однозначним чином ставить у відповідність будь-якого повідомлення допустимої довжини певну перестановку елементів палітри контейнера.

Нижче наведено приклад найпростішого методу перестановки елементів палітри. Нехай повідомлення m - ціле число від 0 n! - 1, де n - число різних елементів палітри. Всі елементи в палітрі впорядковані за зростанням ваги, рівного (65536R + 256G + B). Місця в палітрі, отриманої в результаті приховування, порожні і пронумеровані від 0 n-1. Місце для першого елемента палітри визначається як залишок від ділення m на n [38]. Місце для другого елементу одиничної палітри обчислюється шляхом ділення без остачі m на n і знаходження залишку від ділення отриманого результату на n-1. Таким же чином обчислюються позиції решти елементів палітри, і виходить нова палітра, що відповідає вихідному повідомленню m. Після отримання нової палітри необхідно змінити відповідним чином значення всіх точок зображення.

Нехай палітра складається з трьох елементів, і вони впорядковані в алфавітному порядку: a, b, c. У якості повідомлення використовується максимально можливе m = 3! - 1 = 5. Залишок від ділення 5 на 3 дорівнює 2, тобто в новій палітрі елемент a стоятиме на останній позиції. Далі, $\left[\frac{5}{3}\right] \; mod \; (3-1) = 1 \; mod \; 2 = 1$. Отже, b

залишиться на своєму місці. Очевидно, що с змушене зайняти єдине порожнє перше місце. Таким чином, після приховування палітра набирає вигляду: cba.

Витяг повідомлення відбувається в зворотному порядку. Перший елемент одиничної палітри a займає останнє місце, номер якого дорівнює 2, отже, залишок від ділення m на 3 дорівнює 2 і m не дорівнює 0. Другий елемент одиничної палітри b займає місце з номером 1, значить, залишок від ділення m на 2 дорівнює 1. Таким чином, m = 1 + 1 - 2 + 3 = 5.

Форматні методи приховування в графічних зображеннях

Форматні методи приховування ґрунтуються на особливостях формату зберігання графічних даних. Якщо проаналізувати призначення і зміст полів формату з метою пошуку таких полів, зміна яких не впливає на якість зображення, наприклад, зарезервовані поля, які не використовуються (або не повністю використовуються) в даний час, то в них можна розмістити приховану інформацію. Це найбільш простий метод приховування, але недоліком є те, що всі такі включення легко виявити, знаючи справжнє призначення і типовий зміст таких полів, передбачений стандартом даного формату. Тому для форматних методів, як правило, виявляється можливим побудова повністю автоматизованого алгоритму для виявлення факту приховування.

2.2.2 Форматні методи приховування в файлах BMP

Метод дописування даних в кінець BMP-файлу.

Є найпростішим співвідношенням сторін шляхом приховування, що використовують той факт, що всі стандартні програми визначають кінець даних зображення виходячи із заголовка зображення, який зберігається через підрядник знизу-вгору. Його модифікацією є метод приховування даних після палітри. Він заснований на тому, що початок даних визначається за допомогою значення поля «зсув даних» (навіть у зображеннях без палітри), значення якого можна штучним чином збільшити, а отриману таким чином ділянку BMP-файлу використовувати для приховування повідомлення.

У випадках, коли в BMP-файл зберігається 16-бітове зображення без стиснення, для приховування можна скористатися фактом, що колірні інтенсивності RGB в цьому режимі кодуються за допомогою 5 біт на канал. В результаті старший біт кожного 16-бітного відліку не містить інформацію про колір і може бути використаний для приховування.

Метод приховування в палітрі.

Даний метод заснований на тому, що кожен елемент палітри складається з чотирьох байт, перші три з яких використовуються для кодування кольору, а останній зазвичай дорівнює 0 і не використовується. Таким способом можливо приховати не більше 256 байтів, не змінивши розмір вихідного BMP-файлу.

Тема BMP-файлу містить чотири байти, які дорівнюють 0 і поки не використовуються в форматі, їх використання для приховування також не призводить до збільшення розмірів контейнера. Крім того, довжина будь-якої байтової послідовності, що кодує горизонтальну лінію пікселів зображення повинна бути кратною 4. В разі якщо це не виконано, вона доповнюється нульовими байтами до розміру, кратного 4. На цю особливість формату BMP базується метод приховування в нульових байтів.

2.3 Методи вбудовування інформації в зображення

Мультимедійні об'єкти, як правило, володіють великою надмірністю, що враховується при застосуванні стеганографічних методів. Різна надмірність використовується для побудови різних методів впровадження інформації в зображення.

2.3.1 Група методів заміни в просторової області

Ці методи засновані на принципі заміни біт надлишкової і малозначимої інформації зображення на біти впроваджуваного повідомлення. Найпоширеніший метод цієї групи – це метод заміни найменш значущих бітів послідовно розташованих пікселів зображення бітами впроваджуваної інформації. Найчастіше довжина біт впроваджуваної інформації менше кількості біт зображення, тому після

впровадження з'являються дві області з різними статистичними властивостями, що легко розпізнається статистичними тестами. Тому інформацію, що впроваджується доповнюють інформаційним сміттям – випадковими бітами, щоб її бітова довжина дорівнювала кількості пікселів в зображенні, що використовується для впровадження. Простота реалізації методу і висока корисна ємність контейнера є безсумнівними перевагами методу, однак, при будь-якому спотворенні контейнера вбудована інформація також спотворюється. Щоб визначити корисну ємність контейнера при використанні методу заміни найменш значущого біта, необхідно скористатися формулою [39]:

$$Q = H * W * V * D$$

де H - це висота зображення в пікселях, W - це ширина зображення в пікселях, V - це число компонент кольору, D - це кількість найменш значущих біт в кожній компоненті, Q - це ємність контейнера, яка вимірюється в бітах.

Існує метод випадкового інтервалу, в якому біти впроваджуваної інформації розподіляються по зображенню так, щоб відстань між двома вбудованими бstами були визначені псевдовипадково.

Методом блочного приховування є метод, в якому вихідне зображення розбивається на непересічні блоки обраного розміру. Розмір блоків вибирається так, щоб оптимально впровадити секретну інформацію. Чим більше розмір блоків, тим менше інформації можна впровадити. У кожного з блоків обчислюється біт парності. Також в кожен блок ховається по одному біту впроваджуваної інформації таким чином, що якщо біт парності не дорівнює секретному біту, то один з найменш значущих біт в блоці інвертується для рівності біта парності і секретного біта.

Таким чином зміна контейнера буде мінімальною. До переваг можна віднести можливість модифікації в блоці такого пікселя, зміна якого призведе до найменшої модифікації статистики контейнера, можливість вибору розміру блоку, від якого залежить ступінь спотворення контейнера. Але у цього методу мала стійкість до спотворень контейнера.

Для впровадження даних також використовується метод заміни палітри, яка існує в форматі зображення. Палітра з кількості N кольорів існує як пара значень:

індекс і його вектор кольоровості. Ця пара присвоюється кожному пікселю. Різні послідовності зберігання кольорів в палітрі кодують інформацію, що впроваджується. Таких послідовностей буде N! в палітрі з N кольорів. Такий метод дуже просто реалізувати, він підходить тільки для невеликих за розміром повідомлень і нестійкий до стеганоатак, пов'язаних з модифікацією палітри зображення.

Метод зміни яскравості також приховує інформацію в просторовій області зображення. У контейнера створюється маска такої ж розмірності з елементами нуль і одиниця, розташованими псевдовипадково. Контейнер розділяється на матриці розміром вісім на вісім пікселів, кожна матриця ділиться на два масиви B 1 і B 2, і обчислюється середнє значення яскравості λ 1 і λ 2 для кожного масиву. Порядок вбудовування біта повідомлення проводиться згідно з наступною формулою [40]:

$$S(x,y) = \begin{cases} 1, \text{при } \lambda_1 - \lambda_2 > \text{E} \\ 0, \text{при } \lambda_1 - \lambda_2 < -\text{E} \end{cases} \quad (2.1)$$

де E - це параметр, який відповідає за значення порогу, що є необхідною різницею між середніми значеннями яскравості.

Якщо умова (2.1) не виконується, то одне із значень яскравості λ 1 або λ 2 модифікується, щоб умова виконувалася. Під час вилучення впровадженого біта секретного сполучення між середніми значеннями яскравості в блоці обчислюється різниця, значення якої дозволяє визначити значення впровадженого біта. До переваг такого методу можна віднести той факт, що повідомлення залишається стійким при JPEG-стиску контейнера з невеликими коефіцієнтами стиснення. Однак, використання такого методу не дозволяє впроваджувати багато інформації в контейнер, до того ж при використанні великого порогового значення E стає помітно візуальне спотворення зображення.

Більш складні алгоритми методу зміни яскравості розглянуті в роботах Куттера (M. Kutter), Джордана (F. Jordan), Боси (F. Bossen) і Дармстедтера (V. Darmstaedter), Делейгла (J. - F. Delaigle), Квісквотера (JJ Quisquater).

2.3.2 Група методів приховування в частотній області

Методи заміни нестійкі до стиснення з втратами, яке практично повністю спотворює всю впроваджену інформацію, чого не можна сказати про методи приховування в частотній області. Найбільше поширення в стеганографії отримали такі ортогональні перетворення, як дискретно-косинусне перетворення, швидке перетворення Фур'є і вейвлет-перетворення, що пояснюється використанням їх в алгоритмах стиснення зображень. Такі перетворення можуть бути застосовані до всього контейнера або тільки до його частини.

Існує дуже багато методів, заснованих на одному з перерахованих вище ортогональних перетворень, однак, для використання одного з них при впровадженні інформації в зображення необхідно враховувати, якому стиску воно можливо буде піддаватися зі часом. Так, наприклад, алгоритм дискретно-косинусного перетворення є базовим для стандарту JPEG, тоді як вайвлет-перетворення використовується в JPEG2000.

Метод відносної заміни величин коефіцієнтів дискретно-косинусного перетворення, званий також методом Коха-Жао, є найбільш поширеним методом приховування інформації в частотній області зображення. Вибирається певна кількість, за якою буде порівнюватися різниця двох низькочастотних коефіцієнтів блоків дискретно-косинусного перетворення, на які необхідно розділити зображення. Вбудовування нуля або одиниці в блок відбувається зміною значень коефіцієнтів.

Метод Бенгама-Мемон-Ео-Юнга оптимізує метод Коха-Жао таким чином, що модифікуються не всі блоки зображення, а тільки ті, які більше підходять для цього за своїми властивостями, і в кожному такому блоці вибирається три коефіцієнта замість двох. Властивості таких блоків повинні включати в себе відсутність різких переходів яскравості, тому що інакше значення низькочастотних коефіцієнтів виходять занадто великими, і не будуть занадто монотонними, тому що інакше більшість низькочастотних коефіцієнтів дорівнюватимуть нулю. Вибір придатних блоків здійснюється порівнянням з двома параметрами P L і P H, є граничними значеннями відповідно для першої і для другої властивості, таким чином, щоб не

перевищувалося значення P L і не було недосягнення значення P H. Впровадження нуля відбувається тоді, коли третій коефіцієнт менше будь-якого з перших двох, впровадження одиниці відбувається за рахунок збільшення третього коефіцієнта щодо перших двох. Якщо такі перетворення призводять до сильного спотворення блоку, такий блок не використовується. Цей прийом зменшує спотворення контейнера, та повідомлень, що вносяться до нього.

Метод Фрідріх є комбінацією двох алгоритмів впровадження інформації, один з яких полягає у впровадженні інформації в низькочастотні коефіцієнти дискретно-косинусного перетворення, а інший полягає у впровадженні інформації в середньо частотні коефіцієнти дискретно-косинусного перетворення. Використання двох методів в комплексі дозволяє отримати високу стійкість до стеганографічних атак[10].

2.4 Алгоритми реалізації стеганографічних методів

2.4.1 JSteg

JSteg – один з алгоритмів стеганографії для вбудовування даних в зображення JPEG. Алгоритм JSteg був розроблений Дереком Уфамом в 2004 році [41]. Алгоритм пропонує велику місткість стеганографічних повідомлень: приховане повідомлення може займати до 12.8% від загального обсягу зображення (контейнера).

Алгоритм приховування є заміною найменш значущих біт (НЗБ, LSB) в зображеннях JPEG, стислих з втратами. JSteg замінює молодший біт отриманих після квантування частотних коефіцієнтів ДКП на біт секретного повідомлення. Алгоритм стійкий до візуальних атак. Однак статистичні атаки на JSteg надійно визначають наявність прихованого повідомлення. JSteg замінює біти, і, отже, вносить залежність в частоту появи значень в НЗБ. Як показує рисунок 2.1, JSteg впливає на пари частот появи коефіцієнтів JPEG. Припущення для зміненого зображення полягає в тому, що суміжні частоти однакові. Для визначення очікуваного розподілу обчислюється середнє арифметичне. Отримане значення порівнюється з емпіричним розподілом.

Рисунок 2.2 ілюструє статистичну атаку на JSteg стеганограvму (з 50% заповненням контейнера, тобто 7680 байт). Діаграма являє ймовірність впровадження [42]:

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}}\Gamma\left(\frac{k-1}{2}\right)\int_0^{\chi^2} e^{\frac{-t}{2}} t^{\frac{k-1}{2}} dt$$

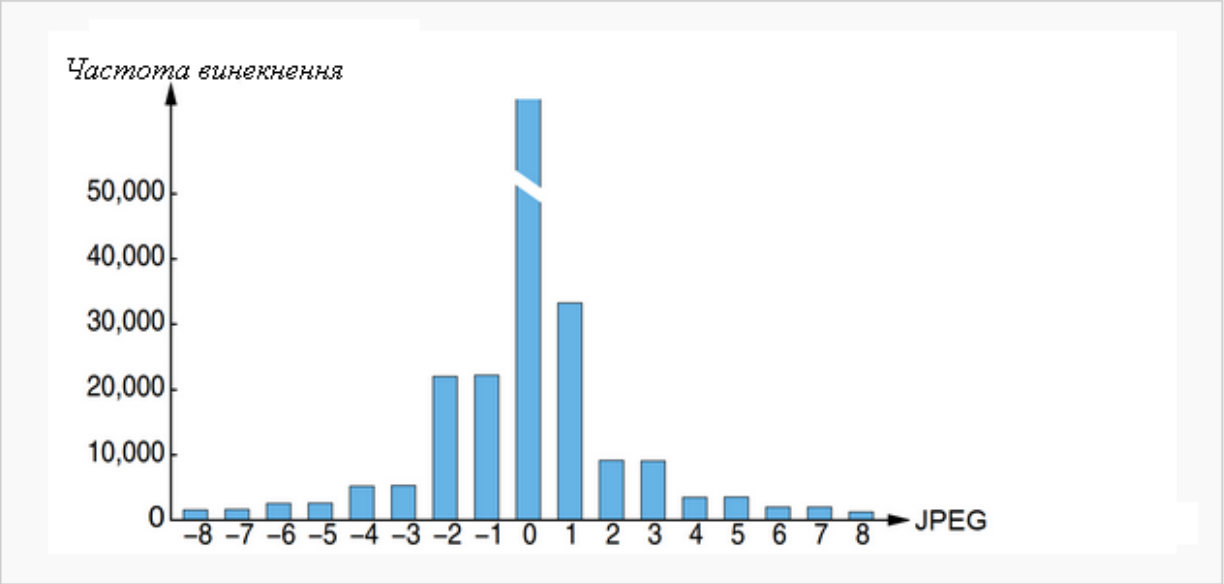

Рисунок 2.1 – JSteg зрівнює пари коефіцієнтів

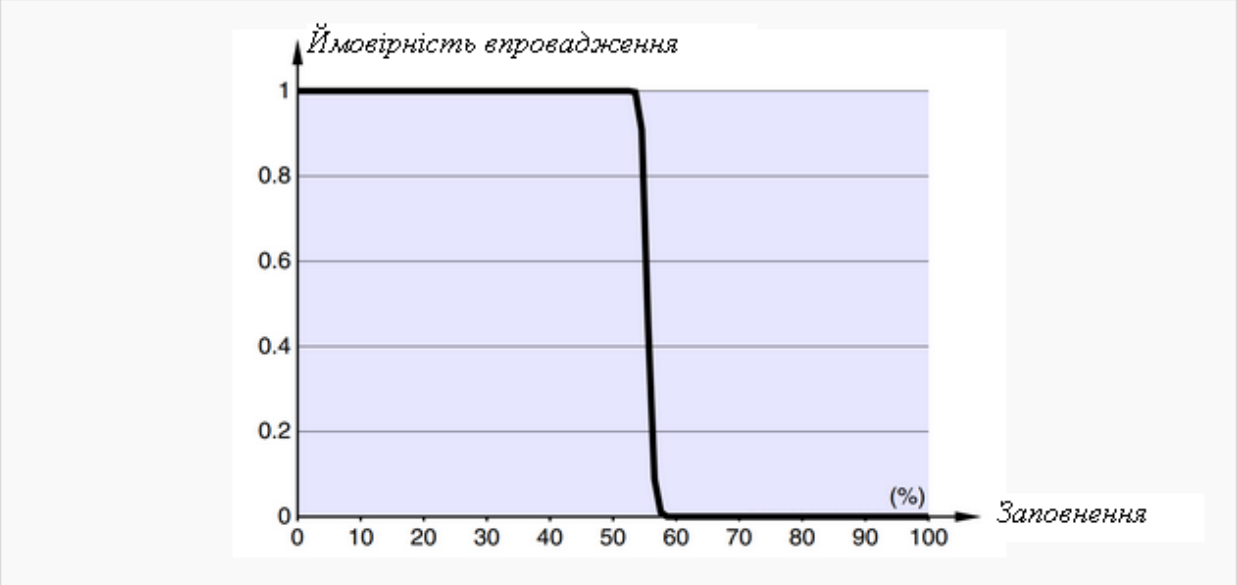

Рисунок 2.2 – Імовірність впровадження в Jsteg стеганограмму (50% заповнення)

2.4.2 Алгоритми стиснення зображень

Цифрове зображення при зберіганні займає великі обсяги пам'яті. Так растрове зображення розміром 1024 на 1024 пікселів з глибиною кольору 24 біт займає 3 Мб. Зрозуміло, що зберігання і передача зображень в такому вигляді є досить трудомістким завданням. Тому завдання представлення зображень в компактній формі (стиснення даних) є досить актуальним. При цьому повинні бути розроблені алгоритми як для кодування, так і для декодування (відновлення) зображень.

Алгоритми стиснення зображень діляться на два великі класи: без втрат і з втратами. У першому випадку в ході компресії інформація про зображення зберігається в повному обсязі, а в другому – частково втрачається. Перша група методів стиснення забезпечує відновлення вихідного зображення без втрат і спотворень. Для зберігання зображень, призначених для подальшої обробки, слід застосовувати формати, що використовують саме такі методи стиснення. Однак, якщо зображення призначене для візуального сприйняття, це не завжди необхідно. У ряді випадків вихідний сигнал вже містить такі спотворення і шуми, що невеликі втрати інформації при кодуванні (на користь високого ступеня стиснення) не зіпсують якості зображення в цілому.

Одна з серйозних проблем комп'ютерної графіки полягає в тому, що до цих пір не знайдений адекватний і однозначний критерій оцінки втрат якості зображення [43]. Для зображень, які спостерігаються візуально, основним є не відрізнятись оком вихідного і компресованого зображення.

Групове стиснення.

Одним з найпростіших методів стиснення зображень є алгоритм RLE (Run Length Encoding - кодування із змінною довжиною рядка). Основною ідеєю цього методу є пошук однакових пікселів в одному рядку. Знайдені ланцюжки однакових елементів замінюються на пари (лічильник повторень, значення), що в певних випадках суттєво зменшує надмірність даних.

Алгоритм в першу чергу розрахований на зображення з великими областями повторюваного кольору (ділова графіка, схеми, малюнки і т.п.). Недоліком такого

підходу є те, що в певних ситуаціях він може замість зменшення призводити до збільшення розміру файлу (наприклад, в деяких випадках при збереженні кольорових фотографій).

Існує багато схем групового стиснення, одну з яких можна проілюструвати наступним чином:

Вхідний потік даних:

17 8 54 0 0 0 97 5 16 0 45 23 0 0 0 0 0 3 67 0 0 8

Потік даних після кодування:

17 8 54 0 3 97 5 16 0 1 45 23 0 5 3 67 0 2 8

Найчастіше для кодування використовується схема, яка називається PackBits. За аналогією зі зберіганням негативних чисел, кожні 7 біт вихідних даних замінюються в результаті на 8 біт. Додатковий дев'ятий біт інтерпретується як прапор стиснення. наприклад:

Вхідні дані: 1,2,3,4,2,2,2,2,4

Дані після кодування: 1,2,3,4,2, & 3,4.

Принцип: Послідовності повторюваних значень кольору замінюються його значенням і кількістю повторень.

Формати: BMP, TIFF, GIF

Коефіцієнт стиснення: 2

Метод Хафмана.

Цей метод названий на честь його розробника (1950). Алгоритм заснований на тому припущенні, що деякі значення сигналу зустрічаються частіше інших. Якщо проаналізувати гістограми зображень, то можна в цьому переконатися. Цей факт можна використовувати для стиснення зображень [44-46]. Використовувати для зберігання значень інтенсивності, які зустрічаються частіше, менше число біт ніж на саме значення. Головна проблема в тому, щоб відокремлювати одне значення від іншого. Адже на різні значення відводиться різна кількість біт.

Вхідний потік даних: CEGADFBEA

Потік даних після кодування: 0010 0001 000011 1 0011 000010 01 0001 1)

Угруповання по байтам: (0010 0001) (000011 1 0) (011 00001) (0 01 00 1 0 1)

Метод стиснення Хафмана можна проілюструвати так, як показано у таблиці 2.2.

Таблиця 2.2 – ілюстрація методу стиснення Хафмана

Значення	Частота згадки	Код Хафмана
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011

Утворені коди унікальні, в сенсі, що можуть бути записані в потік даних без роздільників і маркерів. За кількістю нулів до і після 1 програма відновлення може однозначно визначити значення елемента. Цей метод іноді використовується в ускладненій формі, коли кодується не одне значення, а послідовності значень.

Інша модифікація методу – піддати коди Хафмана груповому стисненню.

Формати: TIFF, GIF

Коефіцієнт стиснення: 3

Метод LZW.

Цей метод названий також в честь його розробників (Lempel, Ziv, Welch). Це універсальний метод, придатний для кодування будь-яких сигналів. Схожий на метод Хафмана, тільки для кодування елементів використовуються коди рівної довжини, а також використовуються коди для послідовностей елементів, які часто зустрічаються.

Складається таблиця всіх кольорів, наявних в стисливому зображенні. Таким чином, замість значення кольору пікселя можна використовувати індекс з таблиці. Найбільш часто зустрічаються кольори на зображенні мають менші індекси, а кольори, що рідко зустрічаються розміщуються в кінці таблиці.

Вхідний потік даних: 123 145 201 4 119 89 243 245 59 11 206 145 201 4 243 245

Потік даних після кодування: 123 256 119 89 257 59 11 206 256 257

Таблиця кольорів (палітра) розміщується між заголовком і власне зображенням. Наприклад так, як показано у таблиці 2.3.

Таблиця 2.3 – палітра кольорів між заголовком і власне зображенням

Індекс	Значення
0000	0
0001	1
0254	254
0255	255
0256	145 201 4
0257	243 245
4095	xxx xxx xxx

Розробники запропонували не тільки спосіб зберігання даних, але і добре документовані алгоритми стиснення і відновлення сигналу. Метод був запатентований, стандартизований і тепер використовується для стиснення будь-якої інформації.

Формати: TIFF, GIF

Коефіцієнт стиснення: 5

Метод LZW, як і RLE, краще діє на зображеннях, що містять однорідні, вільні від шуму ділянки кольорів. При цьому він діє набагато краще, ніж RLE, при стисненні довільних графічних даних, але процес кодування і розпакування відбувається повільніше.

Метод JPEG.

Серед методів стиснення з втратами слід виділити сімейство JPEG, розроблене організацією Joint Photographers Experts Group. Метод заснований на частотних уявленнях зображення і наступних припущеннях [47]. Якщо до сигналу застосувати інтегральне перетворення (Фур'є наприклад), то в результаті в частотному поданні основну, несуть низькі частоти. Високі частоти описують шум і несуттєві деталі.

Видалення 50% високочастотної інформації спричинить за собою видалення 5% корисної інформації, яка міститься в зображенні.

JPEG-стиснення починається з розбиття зображення на квадратні області розміром 8 на 8 пікселів (64 пікселя - 64 байта). Ці області обробляються незалежно. Після перетворення в кожній групі залишається від 2 до 20 байт. При відновленні сигналу повинна бути виконана апроксимація і відновлена вихідна область 8 на 8 пікселів.

Для стиснутого представлення сигналу можуть використовуватися різні перетворення. Найбільш адекватний (якісний) результат дає перетворення Karhunen-Loeve, але воно складно і трудомістке в реалізації. Перетворення Фур'є просто, але це не дає бажаного результату при відновленні. Найбільш придатним виявилося дискретне косинусне перетворення (Discrete Cosine Transform, DCT). При використанні DCT не потрібно працювати з комплексними числами (вихідний сигнал і його спектр речові).

В результаті перетворення вибірки 8 на 8 отримуємо вибірку спектра 8 на 8. При цьому низькі частоти містяться у верхньому лівому кутку спектра, а найвищі в правому нижньому. Високі частоти можна обнулити і не зберігати.

Якщо уявити спектр у вигляді, як показано на рисунку 2.3:

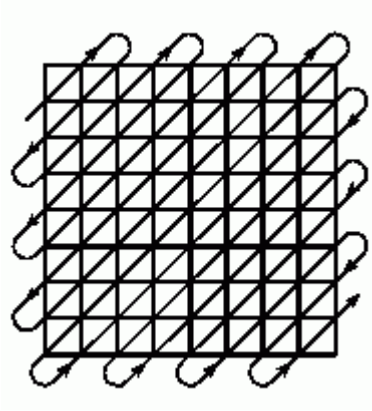

Рисунок 2.3 – Спектр «зиг-заг»

то можна її закодувати за умови використання групового стиснення. На останньому етапі стиснення використовується кодування методом Хафмана для більш ефективного стиснення кінцевих даних.

Принцип: Використовується методика стиснення з втратами. Зберігається не інформація про колір пікселів, а коефіцієнти розкладання по деякому базису.

Формати: JPEG

Коефіцієнт стиснення: в залежності від якості від 10 до 1000.

Позитивними сторонами алгоритму JPEG є те, що користувач може управляти співвідношення розмір / якість, задаючи ступінь стиснення. Вихідне кольорове зображення може позначити глибину кольору 24 біта на точку. За допомогою алгоритму JPEG досягаються великі коефіцієнти стиснення при візуально високій якості зображення. Негативними сторонами алгоритму є те, що при підвищенні ступеня стиснення зображення розпадається на окремі квадратні області (розміром 8x8). Це пов'язано з тим, що відбуваються великі втрати в низьких частотах при квантуванні, і відновити вихідні дані стає неможливо. Крім того може проявлятися так званий ефект Гіббса - ореоли по межах різких переходів кольорів. Крім того, так як це алгоритм стиснення з втратами, зображення оброблені з його застосуванням практично не застосовуються для аналізу і подальшої обробки.

Формат JPEG є дуже популярним форматом, що підтримує компресію. Впровадження інформації для приховування в контейнер, який є зображенням такого формату, має забезпечувати стійкість до стиснення [48-50]. Для цього необхідно при розробці алгоритму впровадження спиратися на основні принципи побудови формату JPEG. Алгоритм стиснення зображень у форматі JPEG працює з YCbCr - спеціальним способом кодування інформації адитивної колірної моделі RGB. Цей спосіб кодування відрізняється тим, що складові колірної моделі визначають не три колірних канали R (red), G (green) і B (Blue), а компоненту Cb, компоненту Cr і компоненту яскравості Y. Тому спочатку складові зображення в колірній моделі RGB переводяться в потрібні величини:

$$Y = R * 0{,}299 + G * 0{,}587 + B * 0{,}114$$

$$Cb = R * (-0{,}169) + G * (-0{,}332) + B * 0{,}5 + 128$$

$$Cr = R * 0{,}5 + G * (-0{,}419) + B * (-0{,}0813) + 128$$

Такий переклад необхідний тому, що більша частина інформації, яка сприймається людським оком, складається з компоненти яскравості Y. З огляду на цей факт, можна вводити так зване проріджування в канали кольоровості Cb і Cr, чим можна значно скоротити обсяг інформації. Наприклад, якщо проріджувати канали Cb і Cr в два рази, 3/4 інформації щодо кольоровості буде втрачено, за рахунок чого обсяг зображення зменшиться в два рази.

Далі отримане для обробки зображення розбивається на сегменти. Ці сегменти представляють собою частини зображення розміром 8 * 8 пікселів. Коли кількість пікселів зображення не ділиться без остачі на такі сегменти, неповні частини доповнюються пікселями, дубльованих сусідніми. Після поділу зображення сегменти, які є робочими матрицями, необхідно піддати дискретно-косинусному перетворенню, яке перетворює їх в матриці частотних коефіцієнтів відповідного розміру.

У матриці дискретно-косинусного перетворення низькочастотні коефіцієнти з'являються в лівому верхньому кутку, а високочастотні – в правому нижньому. Високочастотні коефіцієнти для того, щоб виконати стиснення, відкидаються, так як це не призведе до значного спотворення зображення, тому що система людського зору не чутлива до змін в високочастотній складовій картинці. Щоб це зробити, необхідно провести квантування матриць дискретно-косинусного перетворення. Для початку необхідно створити матрицю квантування, яка є, по суті, матрицею якості. Заповнення цієї матриці відбувається таким чином [51]:

$$Q(i,j) = 1 + ((1 + i + j) * q)$$

де параметр q - це фактор якості, який задається користувачем в діапазоні значень [1, 14].

Чим більше його значення, тим більший ступінь стиснення відбудеться. Кожна з трьох матриць частотних коефіцієнтів, вийшла при дискретно-косинусному перетворенні, ділиться поелементно на матрицю квантування, і елементи матриці, яка виходить в результаті, округлення до найближчого цілого числа. В цій матриці значущі коефіцієнти зосереджені в лівому верхньому кутку, а в правому нижньому – нульові значення, які відкидаються на наступному етапі. Таким чином зникає

високочастотна інформація, яка не сприймає оком, якій відповідали коефіцієнти в правому нижньому кутку матриці дискретно-косинусного перетворення. Квантування має кілька нюансів, які необхідно враховувати, вибираючи значення параметра q, що відповідає за якість. Наприклад, при великих значеннях q кінцеве зображення може бути настільки сильно стиснутим, що все воно розпадеться на одноколірні блоки.

Далі необхідно відкинути непотрібні нулі, отримані в правому нижньому кутку матриці після квантування, тому проведемо зиґзаґоподібний запис з цієї матриці. Він починається в лівому верхньому кутку і закінчується в правому нижньому. Кожен вектор, який вийде після зиґзаґоподібного сканування відповідної матриці, отриманої за допомогою перерахованих вище перетворень, необхідно закодувати алгоритмом групового кодування, який називається RLE (Run Length Encoding).

Кожна послідовність довжиною більше трьох символів, що повторюються в векторі модифікується в три елементи, де перший елемент – це префікс, який позначає нову послідовність символів, що повторюються, другий елемент визначає довжину послідовності повторюваних символів, а третій елемент – це сам повторюваний символ. Елементи в векторі, довжина послідовності яких менше трьох, використовуються в первісному вигляді. Ефективність стиснення вхідного вектора також залежить від вибору префікса. Найоптимальнішим вважається вибір символу префікса таким, який є самим рідкісним у вхідному векторі. Це обумовлено тим, що, коли кодують, наприклад, одиночний символ, який обраний префіксом, його уявлення складається з трьох символів. Якщо в кодуємому векторі занадто багато одиночних входжень такого символу, то після кодування розмір зашифрованих таким чином даних може виявитися більше розміру вихідних даних.

Після того, як здійснено стиснення інформації алгоритмом групового кодування, використовується кодування стислої інформації алгоритмом Хаффмана. Суть цього алгоритму полягає в тому, що чим рідше зустрічається будь-який символ, тим довшим двійковим кодом він шифрується. Кодування може здійснюватися з використанням стандартної фіксованої таблиці, в якій містяться всі символи з відповідними їм двійковими кодами різної довжини, або з використанням своєї

таблиці відповідностей, створеної щодо інформації, яку необхідно закодувати. При кодуванні трійок символів, які виходять після застосування алгоритму RLE, слід враховувати, що більш рідко зустрічаються великі значення коефіцієнтів і довгі послідовності нулів, ніж невеликі коефіцієнти і короткі послідовності нулів.

Після завершення роботи алгоритму Хаффмана, який є кінцевим етапом JPEG-стиснення, створюється двійковий код, який готовий до передачі та зберігання в комп'ютері.

2.4.3 Алгоритм методу LSB

Суть методу заміна найменш значущого біта (Least Significant Bits - LSB) полягає в приховуванні інформації шляхом зміни останніх бітів зображення, які кодують колір на біти приховуваного повідомлення [52]. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини. Принцип приховування інформації показано на рисунку 2.4

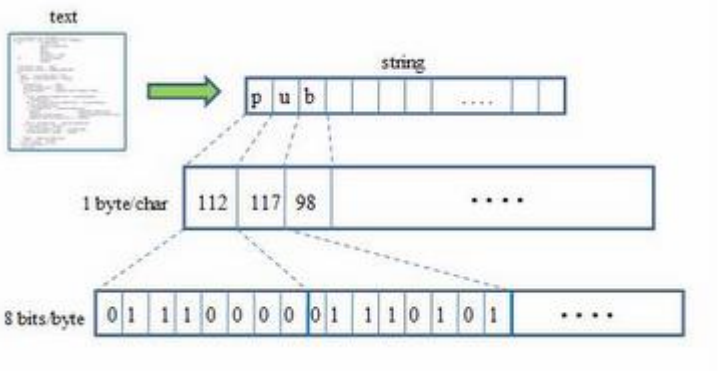

Рисунок 2.4 – Принцип приховування інформації

Перетворення тексту в байтову послідовність.

Як вже описано раніше, в форматі BMP зображення зберігається як матриця значень відтінків кольору для кожної точки зображення, що зберігається. Якщо кожна з компонент простору RGB (їх ще називають каналами кольору) зберігається в одному байті, вона може набувати значень від 0 до 255 включно, що відповідає 24-х бітній глибині кольору. Особливість зору людини полягає в тому, що воно слабо розрізняє незначні коливання кольору. Для 24-х бітного кольору зміна в кожному з

трьох каналів одного найменш значимого біта (тобто крайнього правого) призводить до зміни менш ніж на 1% інтенсивності даної точки, що дозволяє змінювати їх непомітно для ока на свій розсуд [53-55].

Розрахуємо пропускну здатність методу. Якщо відкинути в розрахунках, зазвичай незначне щодо розміру зображення, службову інформацію на початку файлу, то ми маємо можливість потай передати повідомлення розміром в 1/8 розміру контейнера ("розмазати" за останніми бітам в кожному байті матриці кольорів пікселів) або ж розміром в 1 / 4 контейнери (відповідно при використанні 2 останніх бітів в байтах).

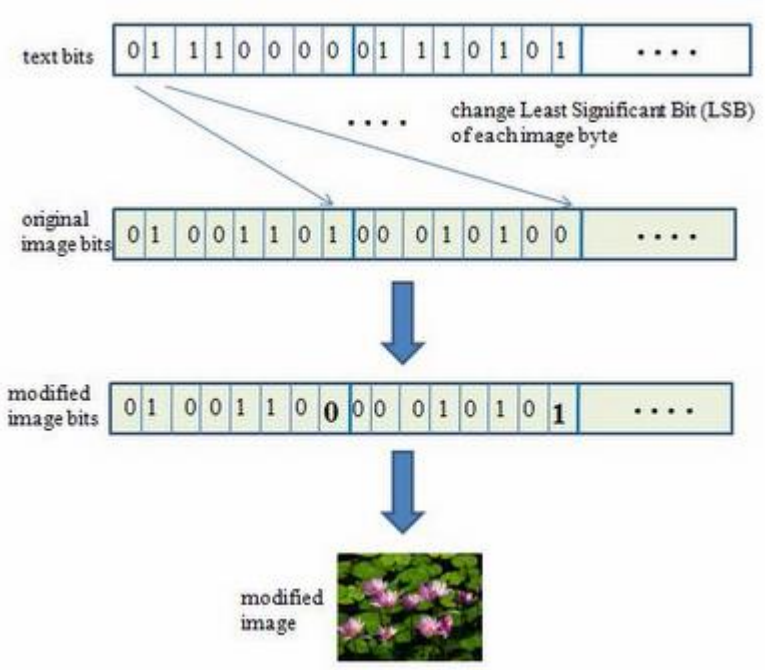

Рисунок 2.5 – Приховування інформації в зображенні

Принцип роботи стеганографічного методу полягає в наступному. Нехай, є 24-х бітове зображення в градаціях сірого. Піксель кодується 3 байтами, і в них розташовані значення каналів RGB. Змінюючи найменш значущий біт ми міняємо значення байта на одиницю. Такі градації, мало того що непомітні для людини, можуть взагалі не відобразитися при використанні низькоякісних пристроїв виведення.

Наведений нижче приклад показує, як повідомлення може бути приховано в перших восьми байтах, що відносяться до трьох пікселів у 24-бітному зображенні:

Пікселі: (00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
A: 01000001
Результат: (0010011 0 1110100 1 1100100 0)
(0010011 0 1100100 0 1110100 0)
(1100100 0 0010011 1 1110100 1)

У прикладі підкреслені тільки біти тільки ті три біта, які були фактично змінені. Застосування стеганографічного методу LSB в середньому вимагає, щоб тільки половина біт зображення-контейнера були змінені.

Невелика модифікація цієї стеганографічної техніки дозволяє використовувати для вбудовування повідомлення два або більш молодших бітів на байт. Це збільшує обсяг прихованої інформації в об'єкті-контейнері, але скритність сильно знижується, що полегшує процес розпізнавання стеганографії. Інші варіації цього методу включають в себе нівелювання статистичних змін в зображенні. Деяке інтелектуальне програмне забезпечення для стеганоаналізу перевіряє області, які складаються з одного суцільного кольору [56]. Для підвищення скритності слід уникнути запису змін в ці пікселі.

Методи LSB є нестійкими до всіх видів атак і можуть бути використані тільки при відсутності шуму в каналі передачі даних. Виявлення LSB-кодованого контейнера здійснюється за аномальними характеристиками розподілу значень діапазону молодших бітів відліків цифрового сигналу.

Висновки до розділу

У рамках другого розділу розкрито теоретичну сторону питання стеганографії, визначено основні формати зображень, запропоновано низку методів здатних здійснити стеганографічний запис. Для подальшого розгляду обрано метод LSB у якості формату зображень застосовано bmp файл.

Суть цього методу полягає в заміні останніх значущих бітів в контейнері (зображення, аудіо або відеозапису) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.

3 РОЗРОБКА АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

3.1 Змістовна постановка задачі

На сьогодні в мережі Інтернет можна знайти безліч безкоштовного або умовно-безкоштовного програмного забезпечення зі стеганографії. Алгоритми, які покладені в основу таких програм, виконують вкраплення конфіденційного повідомлення в так звані контейнери (зображення, аудіо-, відео-). Використання подібних програм дає змогу непомітно для сторонніх осіб передавати по відкритим каналам зв'язку будь-яку закриту інформацію одночасно з відкритою (видимою) інформацією, що не має конфіденційного характеру. Непомітність такої передачі даних може бути використана для реалізації злочинних намірів. Запобігти несанкціонованій передачі інформації методами стеганографії дозволяє стеганоаналіз. Основна задача стеганоаналізу – встановлення факту існування в контейнері прихованої інформації.

Взагалі, виявлення прихованої передачі даних, прихованих одним із багатьох існуючих методів стеганографії в різні формати контейнерів є досить складним процесом. Наприклад, використання широко відомого методу стеганоаналізу на основі критерію хі-квадрат дозволяє отримати гарні результати, якщо вкраплення інформації здійснювалось методом послідовної заміні найменш значущих біт елементів контейнера-зображення або методом вкраплення з заповненням, однак цей метод не спрацьовує коли відбувається псевдовипадковий вибір молодших біт (розподілене вкраплення) [57]. Щоб отримати більш достовірну відповідь про наявність додаткової інформації в потенційному контейнері необхідно мати комплекс стеганоаналітичних методів.

Як відомо, надійність передачі повідомлення в контейнері стрімко падає зі збільшенням розміру повідомлення, яке вкраплюється у контейнер, що досить важко попередити. Це означає, що у випадку перехоплення переповненого контейнера зловмисником, йому буде досить легко виокремити повідомлення з контейнера.

Таким чином проблема полягає у тому, щоб вкрапити якомога більше інформації в контейнер так, щоб навіть у разі перехоплення контейнера, зловмисник не зміг відновити початкове повідомлення.

3.2 Математична модель типової стеганосистеми

Процес звичайного стеганографічного перетворення описується такими залежностями [58]:

$$E : C\times M \rightarrow S; \quad (3.1)$$

$$D : S \rightarrow M, \quad (3.2)$$

де $S = \{(c_1, m_1),(c_2, m_2),...,(c_q, m_q)\} = \{s_1, s_2,...,s_q\}$ – множина заповнених контейнерів (стеганограм). Залежність (1) описує процес приховування інформації, залежність (3.2) – витягування прихованої інформації. Однією із обов'язкових умов при цьому є відсутність «перетину», тобто якщо $m_a \neq m_b$ (причому $m_a, m_b \in M$, а $(c_a, m_a),(c_b, m_b) \in S$), то $E(c_a, m_a) \cap E(c_b, m_b) = \emptyset$. В загальному випадку стеганосистему можна представити як сукупність $\sum(C,M,S,E,D)$ – контейнерів, повідомлень та перетворень, що їх зв'язують. Завжди контейнери с обираються таким чином, щоб заповнений контейнер майже не відрізнявся від порожнього контейнера. Стеганосистема може вважатися надійною, коли $sim[c, E(c,m)] = 1$ (де sim – функція подібності). Контейнер може обиратися двома способами: довільно (сурогатний метод) та підбором найбільш придатного у конкретному випадку контейнера, який зміниться найменше при перетворенні. В останньому випадку контейнер обирається виходячи із умови [59]:

$$c = \max sim[x, E(x,m)]. \quad (3.3)$$

В будь-якому випадку пряме та зворотне перетворення (E та D) мають відповідати одне одному та підлягати умові, що незначне викривлення контейнера (на величину δ) не має призводити до викривлення прихованої інформації [60]:

$$E(c,m) \approx E(c + \delta,m) \text{ або}$$

$$D[E(c,m)] \approx D[E(c + \delta,m)] = m.$$

3.3 Метод найменш значущого біту

Стеганографія – це мистецтво та наука про невидиме спілкування. Це досягається шляхом приховування інформації в іншій інформації, що приховує факт передачі інформації. У стеганографії інформація приховується в зображеннях, аудіо,

відео тощо. Система стеганографії складається з трьох елементів: контейнер (приховує таємне повідомлення), секретне повідомлення та стеганоконтейнер (який є контейнером з вкрапленим повідомлення).

Секретне повідомлення сховане в бітах кольорів пікселя зображення контейнера. Кольоровий малюнок складається з пікселів, які відповідають рівню червоного, зеленого та синього. Червоні, зелені та сині кольори можуть бути об'єднані, щоб створити всі кольори. Кожен колір знаходиться в діапазоні 0-255, і кожен колірний піксель складається з 24 [61].

Вставка найменшого значущого біта – це звичайний, простий підхід до вбудовування інформації в контейнер. Останній біт кожного пікселя підмінюється бітом секретного повідомлення. Під час використання 24-бітового зображення можна використати кожен з компонентів червоного, зеленого та синього кольорів, оскільки кожен із них представлений байтом. Іншими словами, кожний піксель може зберігати 3 біти. Таким чином, зображення 256 × 256 пікселів може зберігати загальну кількість 196 608 бітів вкраплених даних.

Наприклад, сітка 3 пікселів для 24-бітового кольорового зображення може бути такою:

контейнер:

$$10101111\ 00011000\ 11000010\ (175, 24, 194)$$
$$10110000\ 00010110\ 11001010\ (176, 22, 200)$$
$$10100100\ 00011000\ 11000100\ (180, 24, 196)$$

Секретними даними буде ‘a’: ASCII значення ‘a’ 97 = 01100001

Після застосування методу найменш значущого біту стеганоконтейнер виглядатиме

$$1010111\mathbf{0}\ 0001100\mathbf{1}\ 1100001\mathbf{1}\ (174, 25, 195)$$
$$10110000\ 00010110\ 11001010\ (176, 22, 200)$$
$$10100100\ 0001100\mathbf{1}\ 11000100\ (180, 25, 196)$$

3.4 Схема розподілу секрету Шаміра

Ідея, на якій заснована схема Шаміра, полягає в тому, що для інтерполяції многочлена ступеня k-1 потрібно k точок. Якщо відомо меншу кількість точок, то інтерполяція буде неможливою [62-64]. Позначимо: p - велике просте число (більше будь-якого секрету M, який передбачається розділяти в цій схемі). Тоді $M \in Z_p$: n - число часток секрету; k - мінімальний розмір дозволеної групи.

Роботу алгоритму можна розділити на 3 етапи.

Підготовчий етап.

Дилер вибирає випадковим чином коефіцієнти $S_1, S_2, S_3, ..., S_{k-1} \in Z_p$ і складає секретний многочлен:

$$S(x) = S_{k-1}X^{k-1} + S_{k-2}X^{k-2} + ... + S_1X \ + M \bmod p$$

де M – розділяючий секрет, а інші коефіцієнти - довільні елементи поля (коефіцієнти многочлена дилер зберігає в таємниці). очевидно, що S(0) = M. Далі дилер обирає n різних несекретних ненульових елементів $r_1, r_2, r_3, ..., r_n$ із Z_p, кожен з яких ставить у відповідність одному учаснику схеми.

Розподіл секрету.

Дилер обчислює значення наступного многочлена:

$$c_1 = S(r_1), c_2 = S(r_2), ..., c_n = S(r_n)$$

Частка кожного користувача A_i - це пара чисел $(r_i, c_i), i = 1,2,...,n$. Частки роздають учасникам схеми.

Відновлення секрету.

Щоб відновити секрет, треба скористатися інтерполяційною формулою Лагранжа: якщо потрібно побудувати многочлен S(x) ступеня k-1, який при $x_1, x_2, ..., x_k$ приймає відповідно значення $y_1, y_2, ..., y_k$, то цим многочленом буде:

$$S(x) = \sum_{j=0}^{k-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

k як в схемі розподілу секрету многочлен належить обрати так, щоб S (0) = M, то з формули Лагранжа слідує:

$$M = \sum_{i=0}^{k-1} c_i S_i, де: S = \prod_{j \neq i} \frac{r_j}{r_j - r_i}$$

З описаного вище, стає ясно, що для більших значень порога, обчислення стає повільнішим.

3.5 Метод стеганоаналізу «Хі-квадрат»

У методі використовується аналіз гістограми, отриманої за елементами зображення і оцінка розподілу пар значень цієї гістограми. Для BMP файлів пари значень формуються значеннями пікселів зображення, для JPEG – квантованими коефіцієнтами дискретного косинусного перетворення, які відрізняються за молодшим бітом [65].

Молодші біти зображень не є випадковими. Частоти двох сусідніх елементів контейнера мають перебувати досить далеко від значення частоти середнього арифметичного цих елементів. В “пустому” зображенні ситуація, коли частоти елементів зі значеннями 2N і 2N +1 близькі за значенням, зустрічається досить рідко. При вкрапленні інформації дані частоти зближаються або стають рівними.

Ідея атаки χ^2 полягає в пошуку цих близьких значень і підрахунку ймовірності вкраплення на основі того, як близько розташовуються значення частот парних і непарних елементів аналізованого контейнера. Особливістю алгоритму є послідовний аналіз всього зображення і, відповідно, накопичення частот елементів. Метод χ^2 є універсальним, оскільки підходить для аналізу зображень, в які інформація вкраплювалася за допомогою різних стеганографічних програм [66].

Однак результати роботи методу за критерієм χ^2 значною мірою залежать від методу приховання даних. При послідовній заміні НЗБ елементів контейнера і вкрапленні повідомлення з заповненням метод виявляє наявність прихованих даних, як показано на рисунку 3.1, а при псевдовипадковому виборі молодших бітів (розподіленому вкрапленні) метод не спрацьовує, як показано на рисунку 3.2.

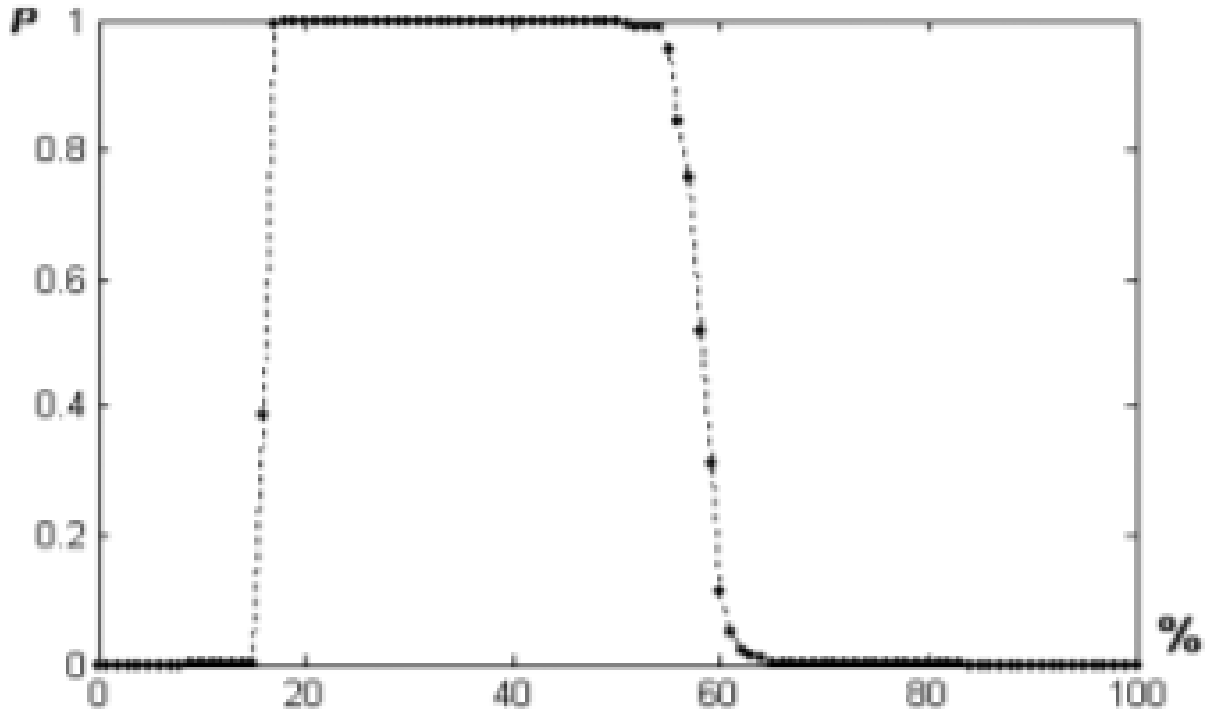

Рисунок 3.1 – Ймовірність послідовного вкраплення за критерієм χ^2

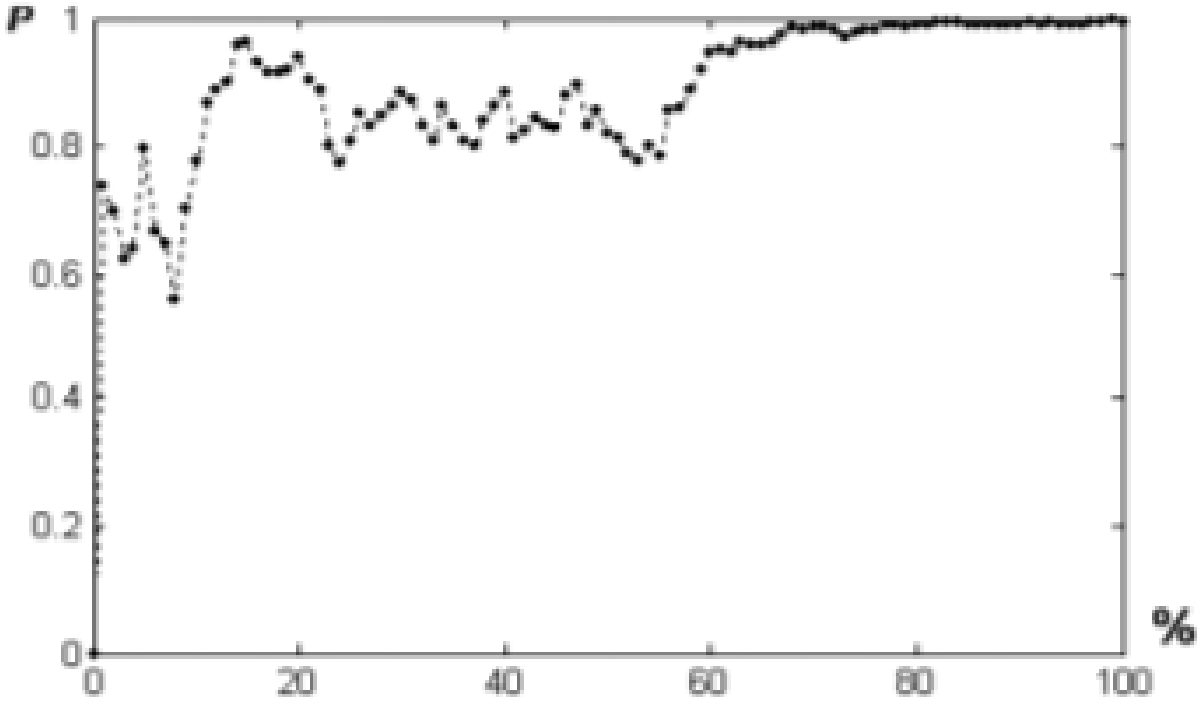

Рисунок 3.2 – Ймовірність вкраплення із заповненням за критерієм χ^2

3.6 Метод стеганоаналізу RS-атака

Одним з оригінальних методів статистичного стегоанализу є метод RS, вперше опублікований в 2001 р. колективом учених під керівництвом Дж. Фрідріх. Скорочення в назві розшифровується як Regular-Singular, тобто «регулярно-сингулярний». Суть методу. Все зображення розбивається на групи по n пікселів $G(x_1, x_2, x_3, ..., x_n)$, де n парне число, наприклад по 2 пікселя, що перебувають поруч по горизонталі. Для групи пікселів визначається функція регулярності або «гладкості» f(G), в якості такої функції можна вибрати, наприклад, дисперсію значень всередині групи, або просто суму перепадів значень суміжних пікселів [67-69]. Під значенням пікселя розуміється ціле число від 0 до 255.

Метод ґрунтується на статистичному припущенні, що для природного зображення, тобто незаповненого контейнера, характерно, що застосування дасть той же розподіл, що й на зображенні, значення пікселів якого зсунуті на одиницю. Для звичайного зображення співвідношення між групами не повинне істотно змінюватися. Значна розбіжність між значеннями свідчить про застосування LSB-стеганографії для молодших біт зображення [70-72].

Розглянемо зміни молодших біт зображення при 100% перезаписі їх бітами повідомлення. Вбудовування випадкового повідомлення довжиною, рівною розміру зображення, призведе до того, що 50% молодших біт будуть інвертовані. На рисунку 3.3 наведена діаграма для типового зображення. На осі абсцис розташована кількість інвертованих біт x, шукана довжина повідомлення p, на осі ординат – відносні значення регулярних і сингулярних груп по відношенню до спільного числа груп зображення.

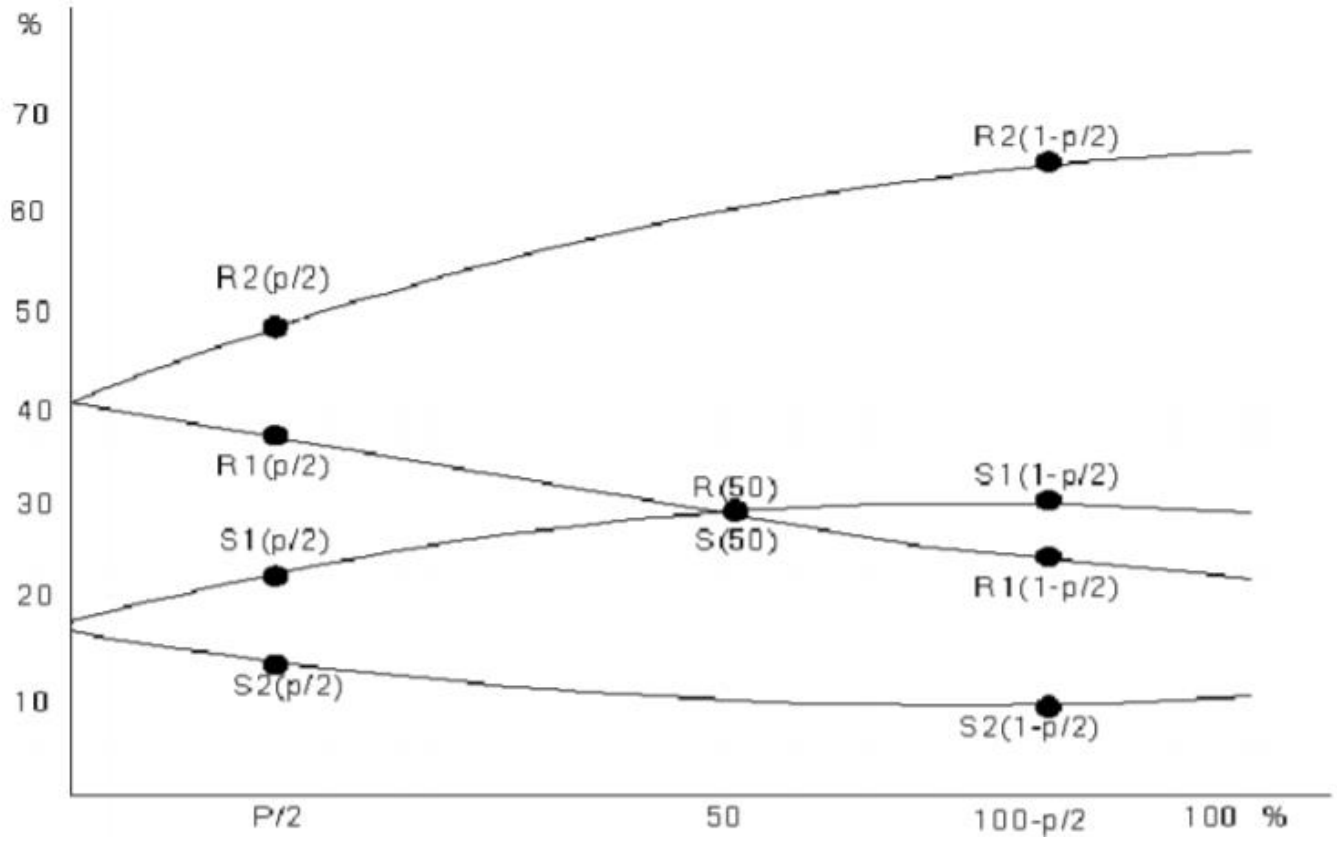

Рисунок 3.3 – RS-діаграма типового зображення

Припускаючи, що в зображення внесене повідомлення довжиною p біт, і при цьому 50% молодших біт, використаних для запису, будуть інвертовані, значення статистик буде одержане у точці p/2. Потім, якщо інвертувати всі молодші біти зображення й перерахувати статистики, на діаграмі вони будуть відповідати точкам кривих при x = 100-p/2. Повній рандомізації молодшої бітової площини відповідає точка 1/2.

3.7 Покроковий опис розробленого алгоритму

КРОК 0. Підготовка вхідних даних, а саме: контейнера, в який буде вкраплено розподіли секрету; ключ для вилучення повідомлення(розподілу) з контейнера; повідомлення, яке необхідно надійно передати; кількість сторін розподілу; кількість сторін розподілу, які необхідні задля відновлення початкового повідомлення із розподілів.

КРОК 1. Виконання розподілу із застосуванням схеми розподілу секрету Шаміра до початкового повідомлення зі вказаними параметрами розподілу: кількість сторін розподілу та кількість сторін для відновлення розподілу.

КРОК 2. Вказання ключа, який контролює вкраплення інформації у контейнер за допомогою генератору псевдовипадкових чисел, що, в свою чергу, впливає на обрання бітів вкраплення та зворотнього вилучення.

КРОК 3. Виконання вкраплення усіх розподілів у копії початкового контейнера та їх подальша передача учасникам.

КРОК 4. Отримання стегоконтейнерів з розподілами у кількості вказаній у параметрі, що відповідав кількості учасників розподілів, завдяки будь-якій вибірці з них у кількості, яку вказано у параметрі, що відповідає за кількість учасників, необхідних для відновлення повідомлення [73-74].

Алгоритм у вигляді блок-схеми зображено у додатку А(Плакат 2).

4 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ

4.1 Засоби розробки

При створенні програмного продукту були використані такі засоби для програмування на C# як MS Visual Studio [2015], генератор випадкових ключів, а також база даних MySQL.

C# проста у використанні, та водночас повноцінна мова програмування, що надає багато засобів для структурування і підтримки великих програм. Вона краще за C обробляє помилки, і, будучи мовою дуже високого рівня, має вбудовані типи даних високого рівня, такі як гнучкі масиви і словники, ефективна реалізація яких на C потребує значних витрат часу. Також для розширення функціональності можна використовувати готові бібліотеки, які отримуються напряму в середу розробки через вбудований у Visual Studio 2017 менеджер пакетів NuGet Package Manager.

C# дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. C# поставляється з великою бібліотекою стандартних модулів, які можна використовувати як основу для нових програм або як приклади при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережними з'єднаннями і навіть інтерфейсами до різних графічних бібліотек.

C# - інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови. Він також зручний як настільний калькулятор. C# дозволяє писати зручні для читання програми. Програми, написані мовою C#, звичайно значно коротші еквівалента на C або C++ з декількох причин:

- типи даних високого рівня дозволять Вам виразити складні операції однією інструкцією;
- наявність новіших методів;
- широкий вибір методів

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників — мов C++, Delphi, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

Контролер (controller) представляє шар класів, що забезпечують зв'язок між користувачем і системою, поданням і сховищем даних. Він отримує дані, що вводяться користувачем і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді подання. Контролер відповідає за визначення моделі подання та отримання даних від подання. Зазвичай на цьому рівні знаходиться вся внутрішня бізнес-логіка застосування.

Подання (view) - це власне візуальна частина або призначений для користувача інтерфейс програми. Подання повинно відображати функціональність моделі у повній мірі, тому що це той рівень, з яким взаємодіє користувач системи.

В якості основної мови програмування для цієї роботи C# був вибраний за рахунок наявності наукових і графічний бібліотек. Інші мови не мають аналогів, окрім Java. C# є крос-платформним і однаково виконується на різних платформах(Windows, UNIX, Macintosh). Водночас він є скриптовою мовою(для розробки сценаріїв), що дозволяє швидко експериментувати і знаходити рішення – це полегшує науково-орієнтовану розробку застосувань.

Об'єднання кращих ідей сучасних мов програмування (Java, C++, VisualBasic та ін.) робить мову C# не просто сумою їх достоїнств, а мовою програмування нового покоління.

Головна вигода від написання Windows-додатків з використанням Windows Forms - це те, що Windows Forms гомогенізують (створюють однорідну (гомогенну) структуру) програмну модель і усувають багато помилок і протиріччя від використання Windows API. Наприклад, кожен досвідчений програміст під Windows знає, що деякі стилі вікна можуть застосовуватися тільки до вікна, коли воно вже

створено. Windows Forms значною мірою усувають таке протиріччя. Якщо ви хочете існуючого вікна задати стиль, який може бути присвоєний тільки в момент створення вікна, то Windows Forms спокійно знищить вікно і знову створить його з вказаним стилем. Крім того,. NET Framework classlibrary набагато багатший, ніж Windows API, і коли ви будете писати програми, використовуючи Windows Forms, ви отримаєте в розпорядження більше можливостей. Написання програми з використанням Windows Forms потребують меншого кількості коду, ніж додатки, які використовують Windows API або MFC.

4.2 Вимоги до технічного забезпечення

Для правильної роботи даного програмного продукту до складу технічних засобів повинні входити:

а) комп'ютер з такою конфігурацією:
   1) процесор з тактовою частотою не нижче 1 ГГц;
   2) дискова підсистема – 40 Гб;
   3) достатній об'єм оперативної пам'яті (не менше 512 МБ);
   4) інші складові можуть мати будь-які параметри, тому що вони не значним чином впливають на роботу програми;

б) додатково має бути встановлене таке програмне забезпечення:
   1) операційна система Windows XP/Vista/Windows7/Windows 10 або ін.;
   2) .Net Framework 3.5 і вище;

в) комп'ютерна периферія, до складу якої входить:
   1) монітор;
   2) мишка;
   3) клавіатура.

Інтерфейсний рівень являє собою рівень відображення даних на стороні клієнта. Дані відображаються користувачеві за допомогою графічного інтерфейсу у вигляді Windows Forms.

4.3 Розробка програмного застосунку

Для зображень формату BMP реалізується метод заміни найменш значущого біта. Такий метод є дуже нестійким до будь-яких спотворень контейнера, а також до його стиснення, яке знищує всю приховану інформацію. До того ж впровадження таким чином інформації легко виявити стеганографічними атаками.

Для більшої міри захисту інформації, прихованої в зображенні методом заміни, можна застосовувати криптографічні методи, які будуть шифрувати повідомлення перед його вкрапленням, а ще, в додаток, застосовано схему розподілу секрету Шаміра, накладаючи при цьому ключ. Таким чином інформація буде захищена чотирма рівнями безпеки, які забезпечать приховування факту передачі повідомлення, а при його виявленні неможливість його розшифровки.

Поряд з криптографією можна розробити і застосувати алгоритми, які забезпечать вибір пікселя для впровадження в нього біт повідомлення згідно якомусь певному правилу. Такий підхід в порівнянні з послідовним вибором пікселів для впровадження інформації забезпечить захист повідомлення від розшифровки навіть при виявленні сторонньої інформації в контейнері, так як біти повідомлення будуть вилучені з зображення у випадковому порядку, визначеним алгоритмом. Така модифікація методу заміни найменш значущого біта застосована в даній роботі за допомогою використання функції, яка генерує випадкове значення порядкових номерів пікселів. Діаграма послідовності зображена у додатку А(Плакат 3).

Після завантаження зображення виконується розрахунок корисної ємності контейнера. Введений користувачем текст представляється в бінарному вигляді згідно кодуванні UTF-8, в якій символи різних алфавітів кодуються різним числом байт. Наприклад, літери російського алфавіту кодуються двома байтами, тобто шістнадцятьма бітами, а букви латинського алфавіту, арабські цифри і розділові знаки кодуються одним байтом.

Далі, зображення, яке є контейнером, розбивається на R, G і B складові. Генерація порядку пікселів для зміни відбувається згідно функції, яка генерує псевдовипадкові числа, які будуть використовуватися як порядкові номери

компонент у векторі, що містить всі компоненти всіх пікселів. Функція генерує псевдовипадкові числа щодо початкового значення, заданого програмою.

Послідовність біт повідомлення впроваджується в псевдовипадковому порядку. У кожній компоненті змінюється один найменш значущий біт, тобто той, якій є найменшим. При впровадженні нуля в нульовий молодший біт і впровадженні одиниці в молодший біт дорівнює одиниці значення компоненти не змінюється. Це враховується при розрахунку відносини пікового сигналу до шуму. Діаграму діяльності зображено у додатку А(Плакат 4).

Ставлення пікового сигналу до шуму обчислюється з використанням підрахованих змінених елементів зображення. Зображення із запровадженою інформацією створюється копіюванням завантаженого зображення для впровадження і внесення в нього необхідних змін. Початкове зображення залишається оригінальним, зображення із впровадженою інформацією зберігається.

Витяг даних зі стеганоконтейнера відбувається тими ж етапами, які застосовуються при впровадженні даних, тільки у зворотному порядку. Записуються компоненти всіх пікселів зображення в порядку слідування R, G, B в один вектор. Обчислюється довжина повідомлення. Це значення записано на початку масиву повідомлення, тому можна його отримати. Після отримання довжини повідомлення можна витягти все повідомлення. Діаграму класів наведено у додатку А(Плакат 5).

4.4 Керівництво користувача

Для запуску програмного застосування необхідно відкрити файл застосунку, який знаходиться у папці з проектом. Відкриється головна форма інтерфейсу застосунку, що зображена на рисунку 4.1.

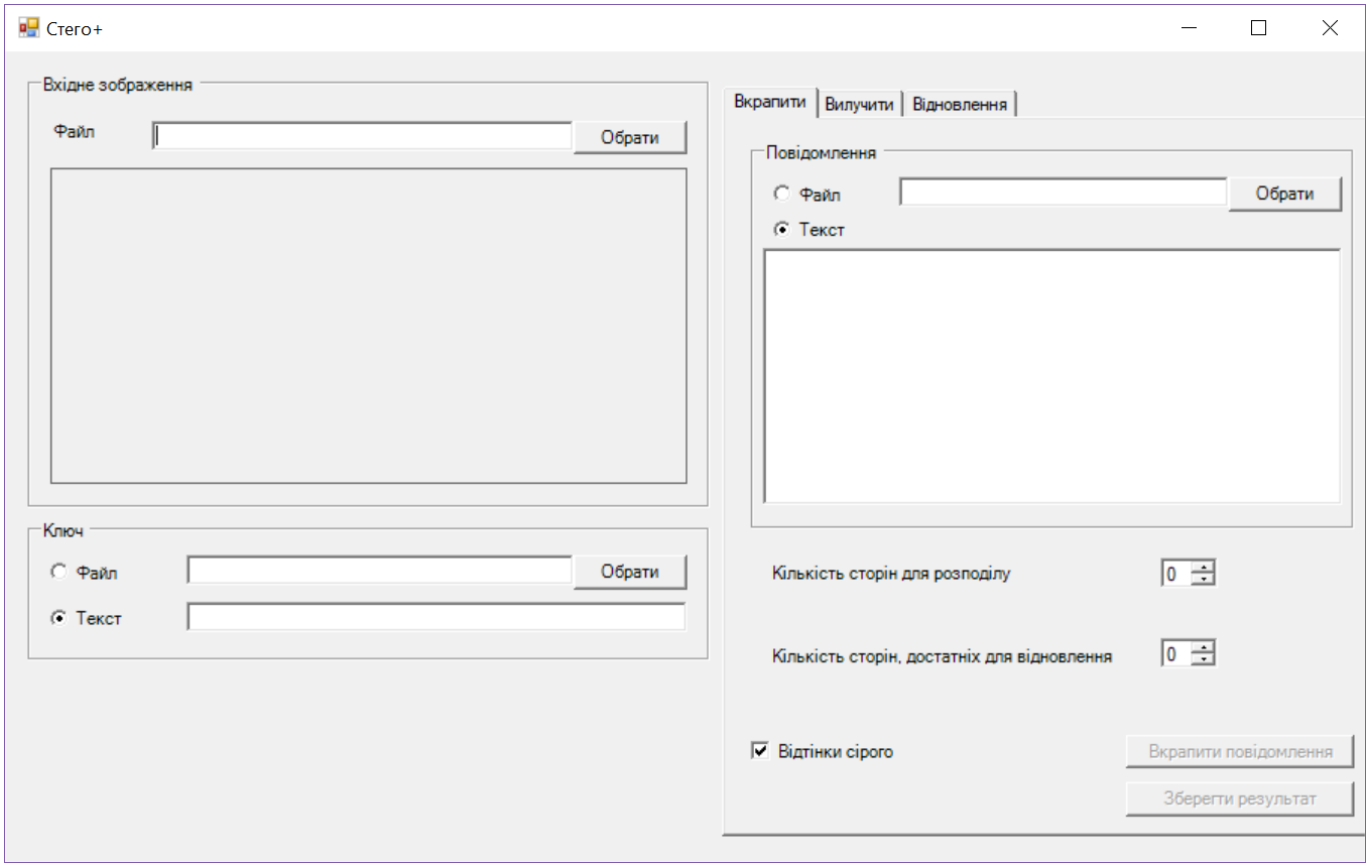

Рисунок 4.1 – Головне вікна програми

З головного вікна існує можливість вкрапити повідомлення з попереднім розподілом за схемою Шаміра, вилучити повідомлення зі стеганоконтейнера за допомогою ключа, відновити початкове повідомлення, за наявності необхідної кількості розподілів секрету без яких неможливе відновлення початкового секрету. Для початку демонстрації роботи, необхідно обрати файл-контейнер для повідомлення, натиснувши кнопку «Обрати». Після обрання контейнеру, він буде зображений у спеціальному місці, як показано на рисунку 4.2.

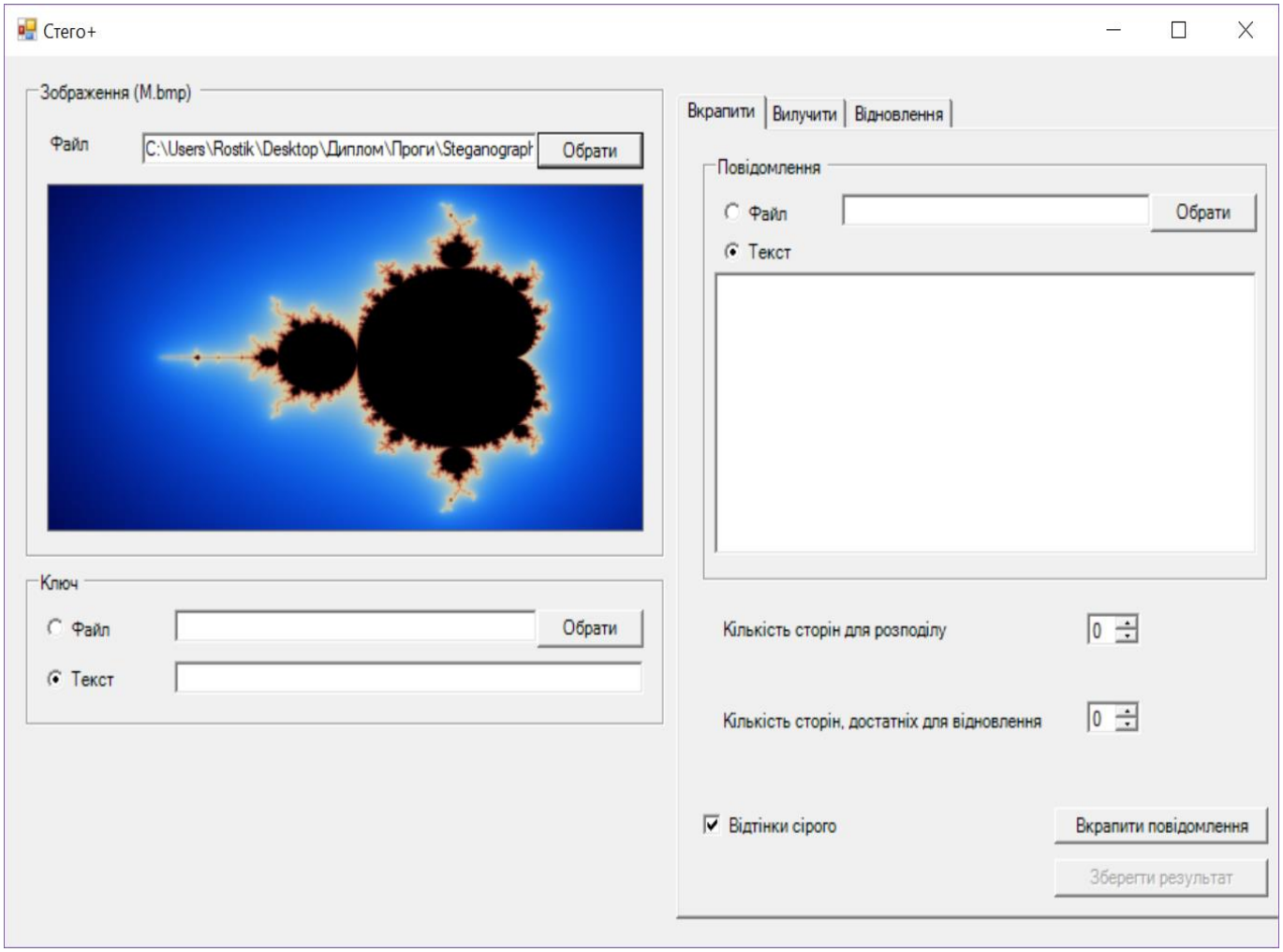

Рисунок 4.2 – Вибір зображення

Для того, щоб задати не послідовне вкраплення в контейнер, необхідно вказати ключ для вказання алгоритму послідовності вкраплення повідомлення. Ключ можна вказати власноруч або обрати з текстового файлу. Цей ключ ускладнює зловмиснику вилучення повідомлення з контейнера у разі перехоплення: невірний ключ надає неправильний порядок символів у повідомлені, що вкраплене. На рисунку 4.3 показано як вказується ключ власноруч.

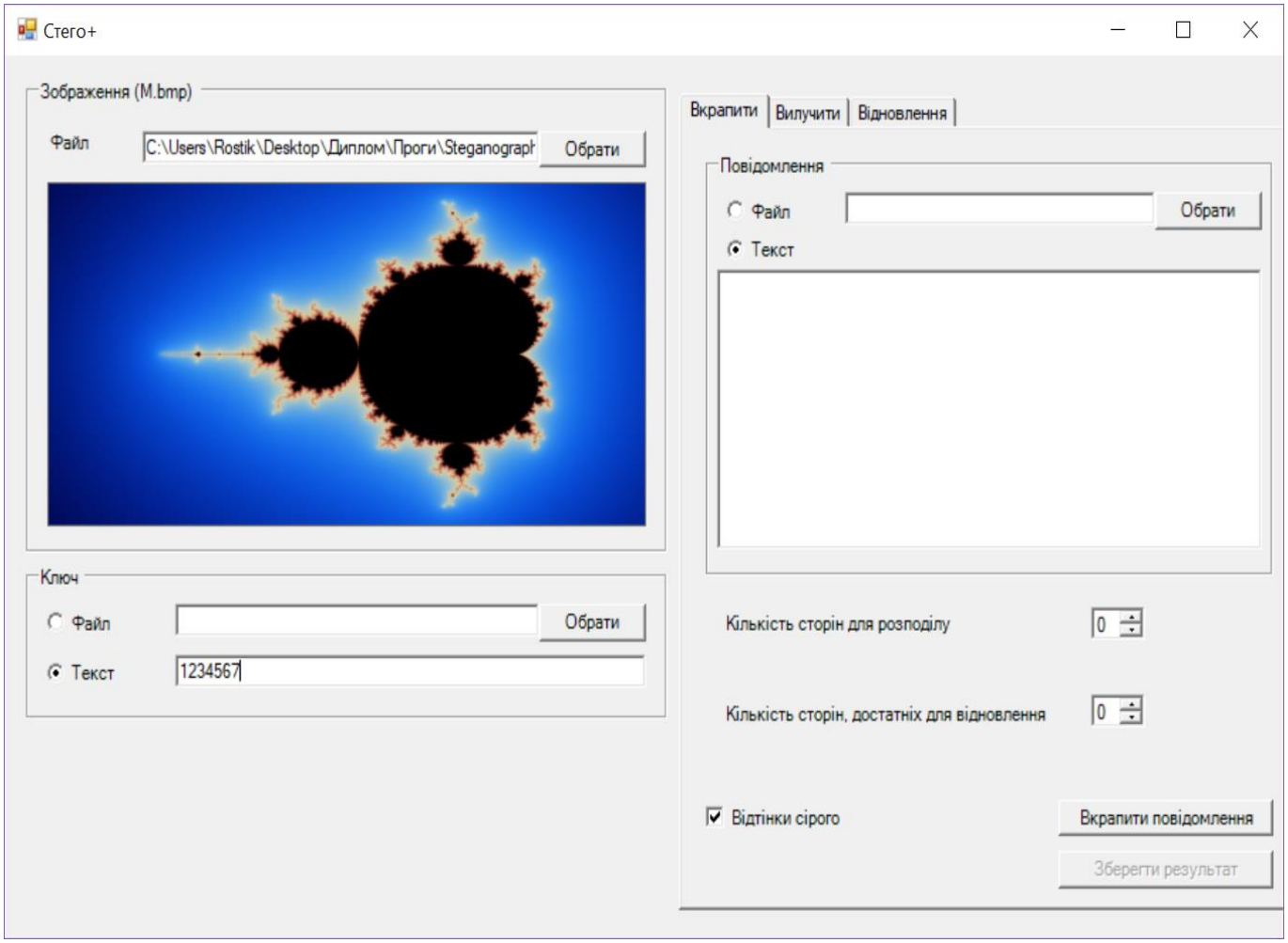

Рисунок 4.3 – Вибір ключа для вкраплення повідомлення

Наступним кроком необхідно вказати текст повідомлення. Це робиться одним із двох можливих способів. Перший спосіб полягає у тому, що користувач власноруч набирає текст повідомлення із клавіатури, в той час як другий спосіб вимагає наявності повідомлення в текстовому файлі формату .txt. Невелике повідомлення зручніше вказати власноруч, в той час як досить великі – просто загрузити із файлу. Також варто звертати увагу на розмір контейнера задля того, щоб приблизно розуміти максимально можливу довжину повідомлення, що буде оброблено алгоритмом та вкраплено в контейнер. Приклад введення повідомлення зображено на рисунку 4.4.

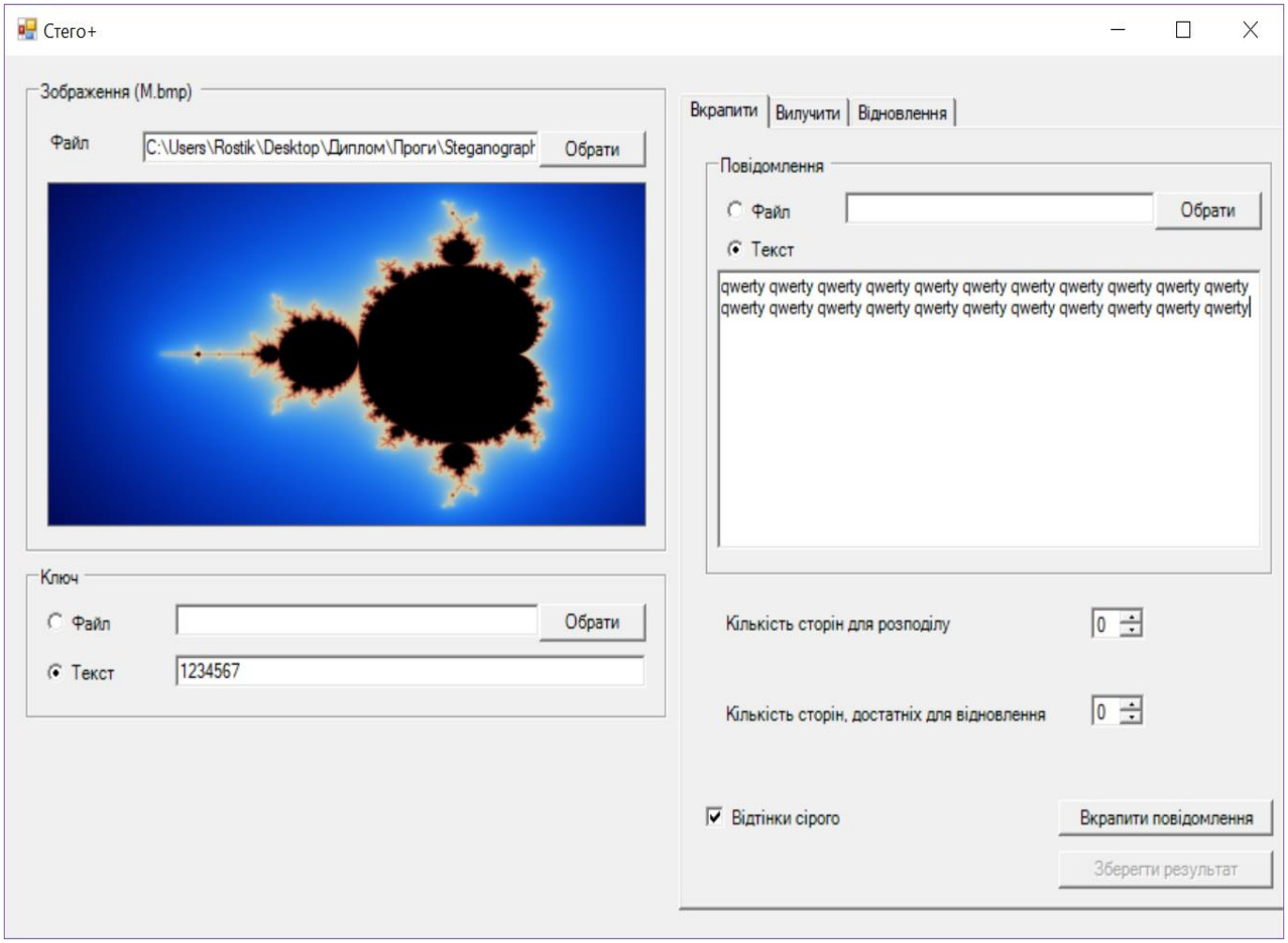

Рисунок 4.4 – Введення повідомлення

Наступний етап – задання параметрів для схеми розподілу секрету Шаміра. Необхідно вказати кількість учасників розподілу та необхідну кількість учасників для відновлення повідомлення. Даний розподіл гарантує невідновлення даних у разі наявності розподілів у меншій кількості, ніж кількість учасників для відновлення, які було вказано перед розподілом у відповідному полі. Завдяки схемі Шаміра, застосунок додає надійності у разі перехоплення зловмисником одного зі стегано контейнерів. Навіть у разі вилучення зловмисником змісту контейнера він отримає лише один з розподілів, який сам по собі ніякої інформації не дає. Приклад заповнення значень параметрів для схеми Шаміра показано на рисунку 4.5.

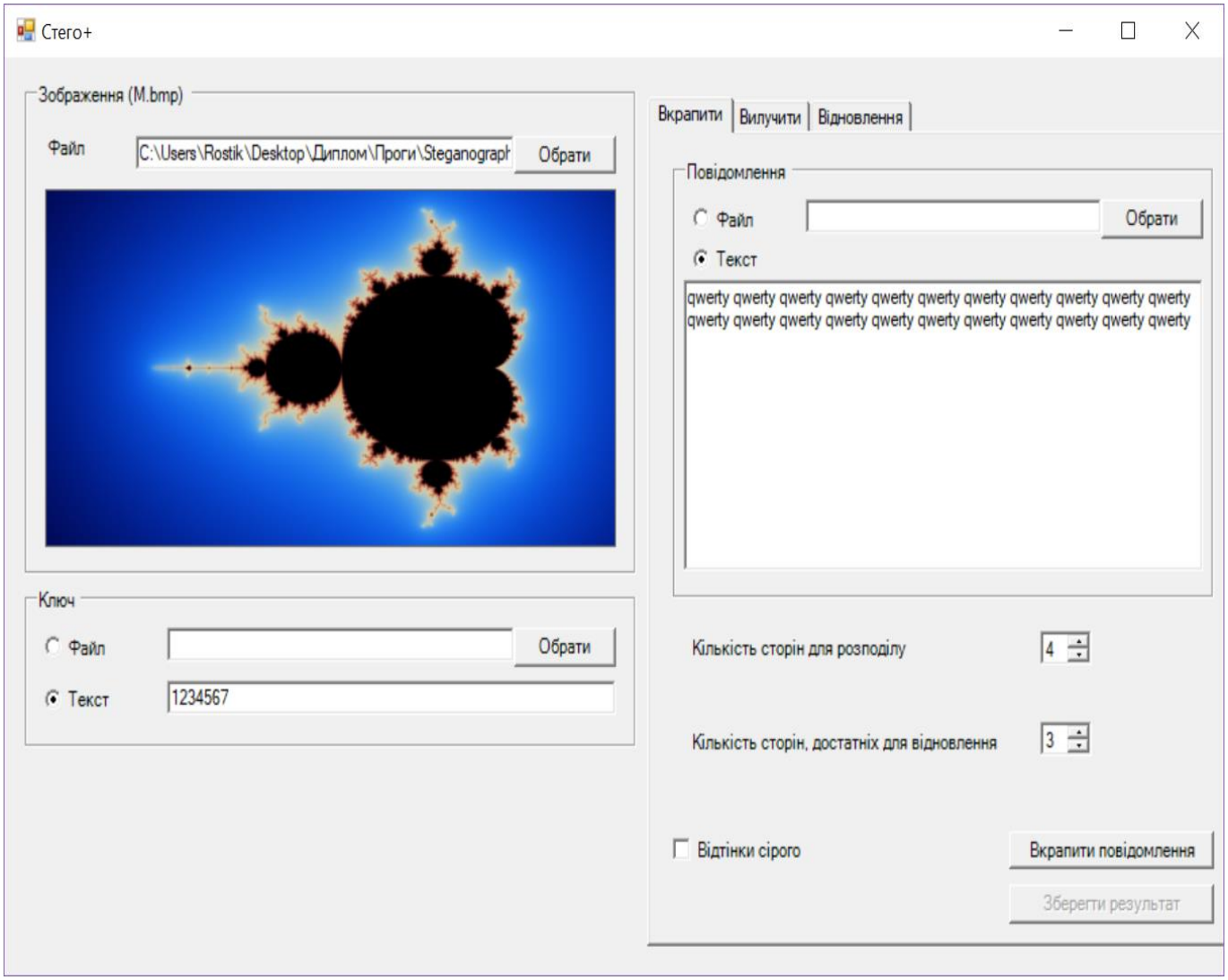

Рисунок 4.5 – Введення параметрів розподілу секрету

Натиснувши кнопку «Вкрапити повідомлення», відбувається вкраплення повідомлення в контейнери з додатковим накладанням ключа, після обробки схемою розподілу секрету Шаміра. Результат роботи алгоритму видає файли, які є заповненими контейнерами, що містять розподіл. Файли зберігаються у кількості, яку вказано у кількості розподілів. Результат роботи застосунку продемонстровано на рисунку 4.6, де показано файли, названі цифрами, в яких, власне, і знаходяться розподіли.

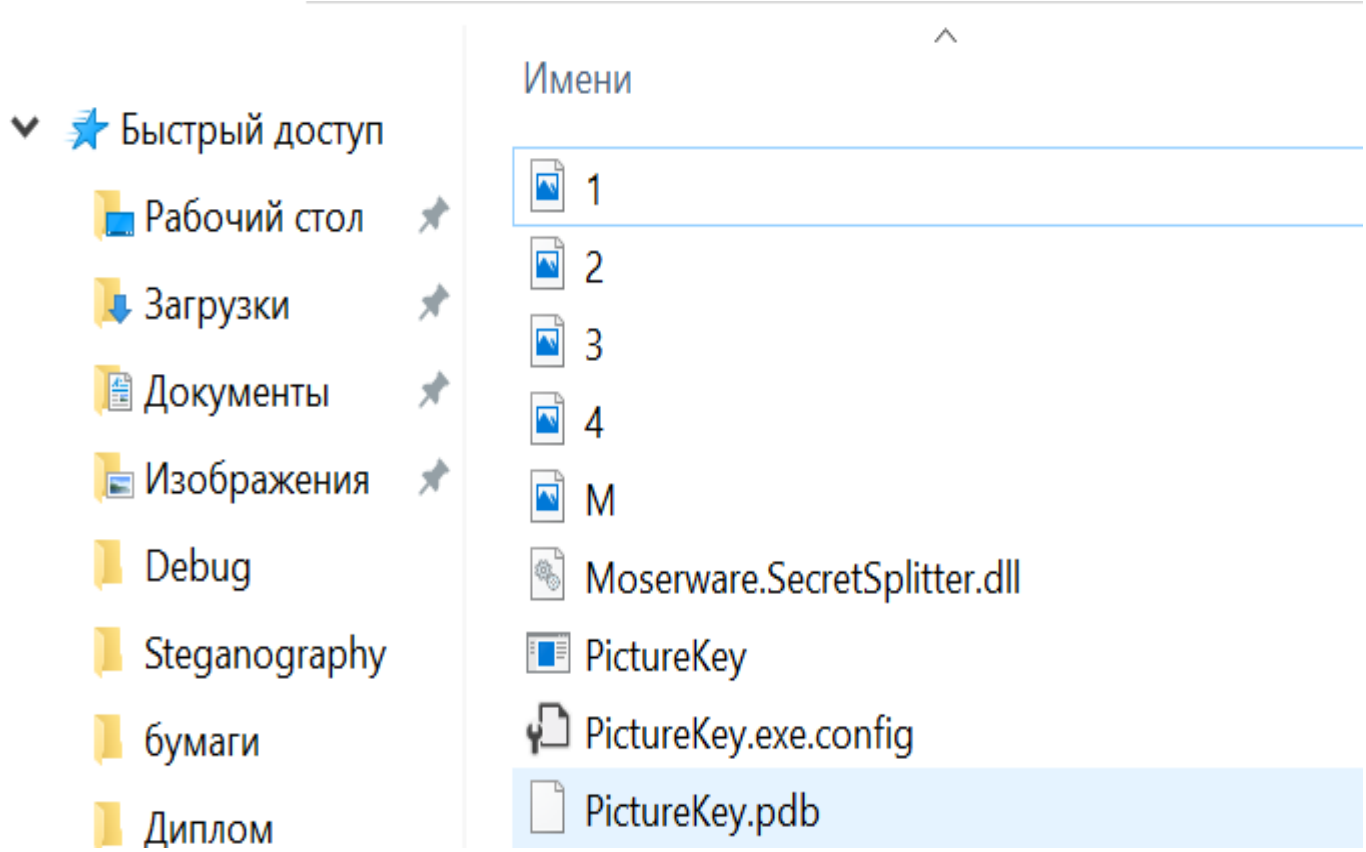

Рисунок 4.6 – Результат роботи застосунку

Тепер здійснимо вилучення щойновкраплених розподілів зі стеганоконтейнерів за допомогою ключа, вказаного при вкрапленні. Для цього необхідно завантажити стеганоконтейнер із вкрапленням, вказати ключ, що було встановлено при вкрапленні, та перейти на вкладку «Вилучити». Як і згадувалось, інформації, яка б хоч трохи нагадувала початкове повідомлення, тут не знаходиться. Вигляд кожного розподілу секрету неінформативний для зловмисник, а також гарантує невідновлення початкового повідомлення. Оскільки в результаті роботи застосунку в режимі вкраплення повідомлення було встановлено чотири розподіли, то нижче буде показано вміст цих чотирьох контейнерів. Вміст контейнерів показано на рисунках 4.7 – 4.10.

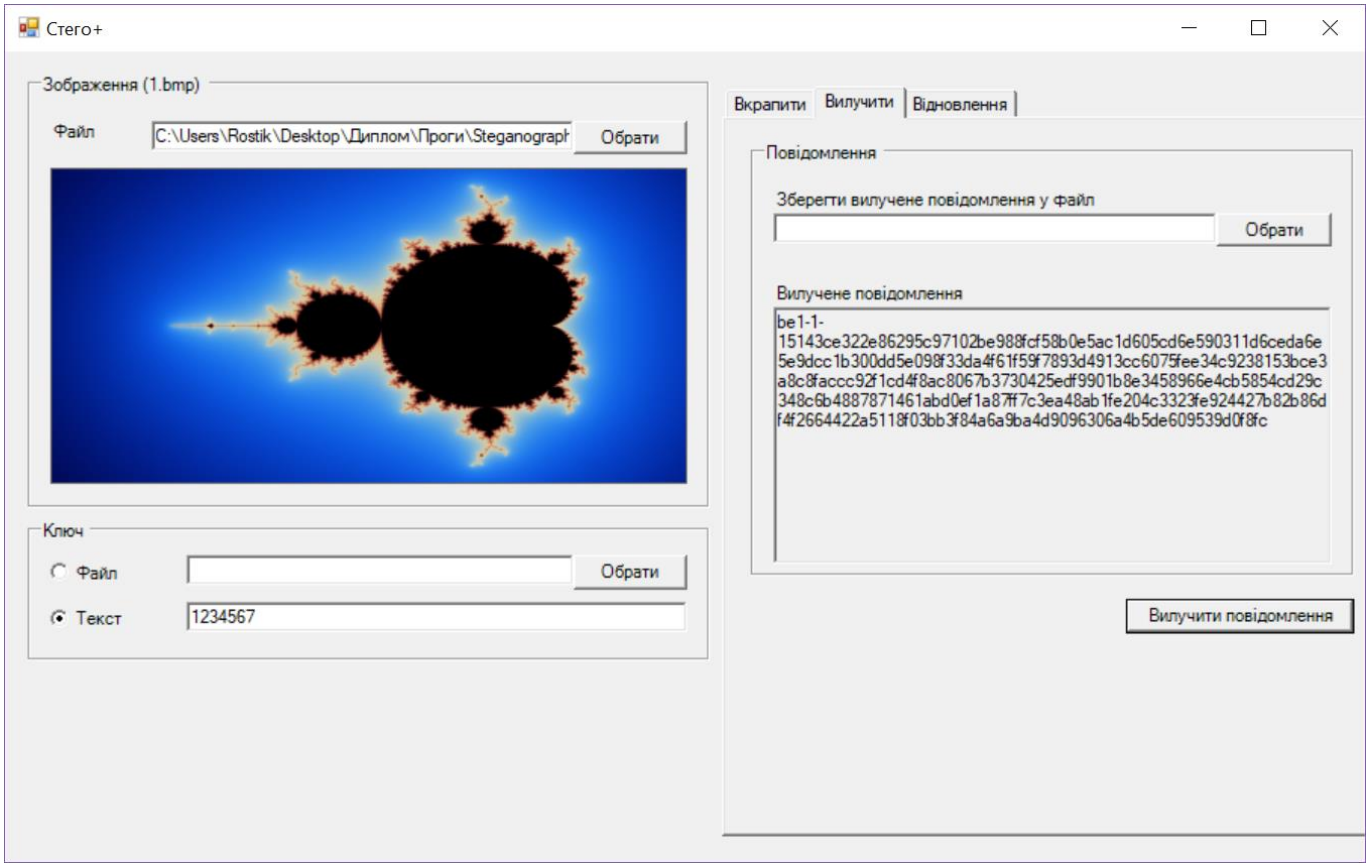

Рисунок 4.7 – Зміст першого контейнера після вилучення даних за ключем

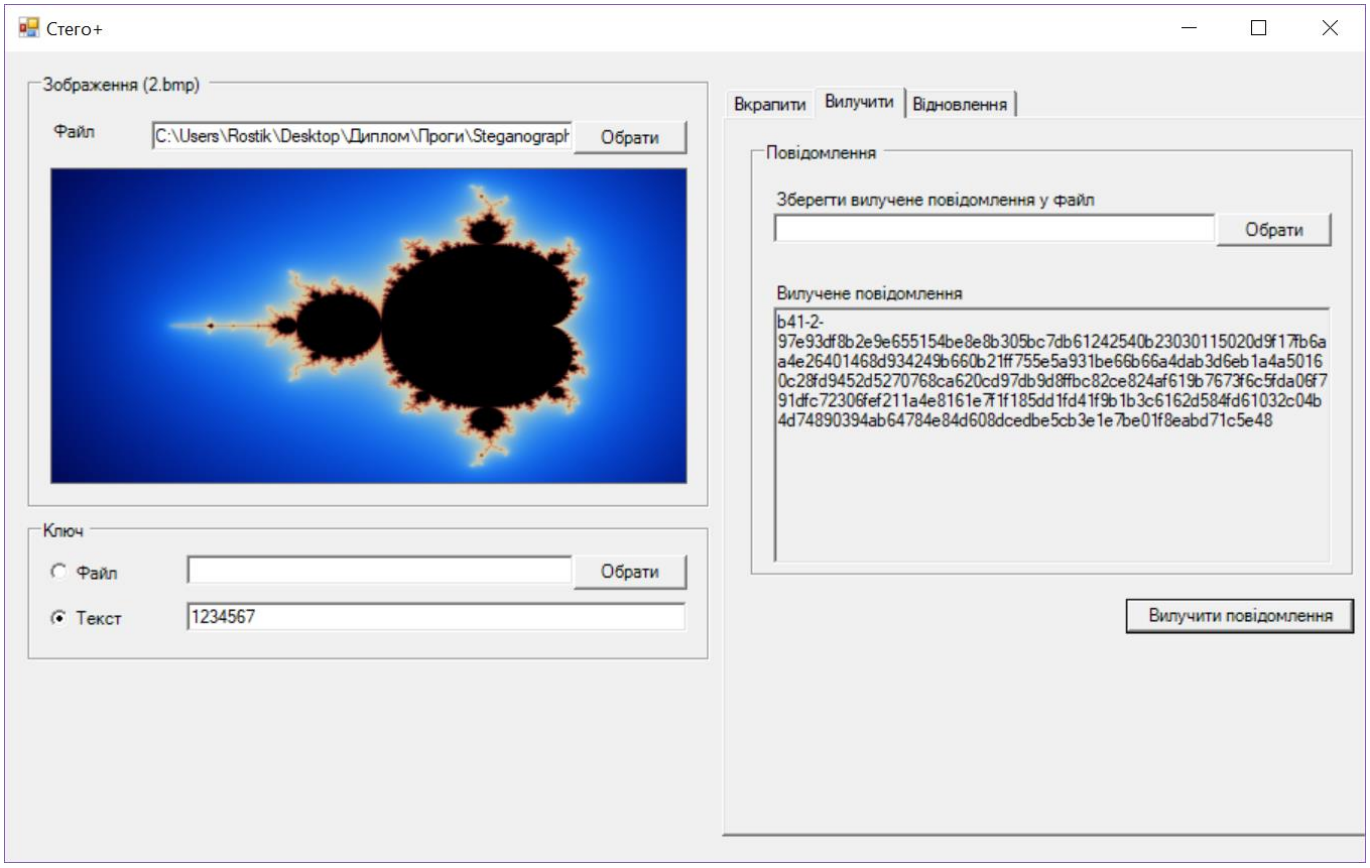

Рисунок 4.8 – Зміст другого контейнера після вилучення даних за ключем

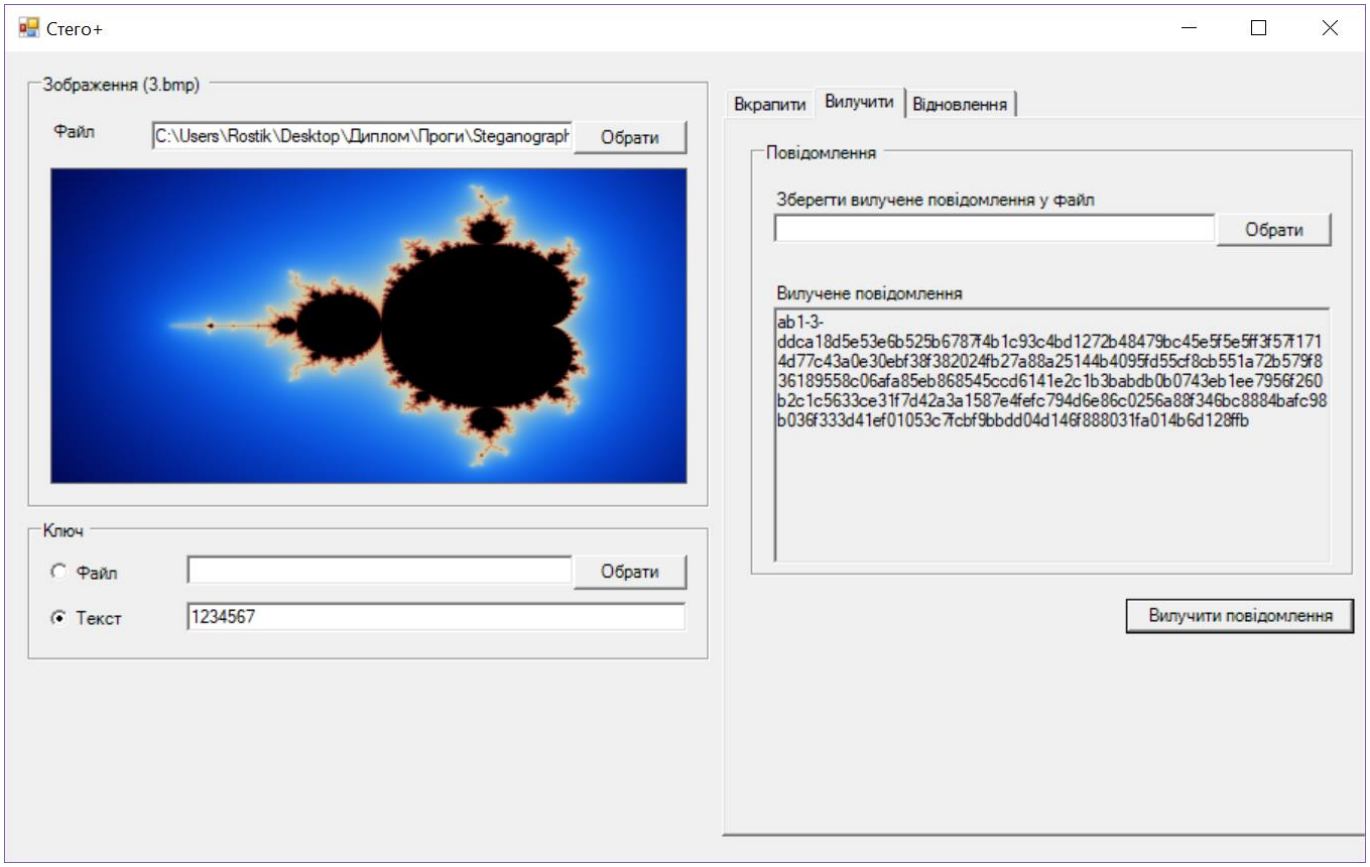

Рисунок 4.9 – Зміст третього контейнера після вилучення даних за ключем

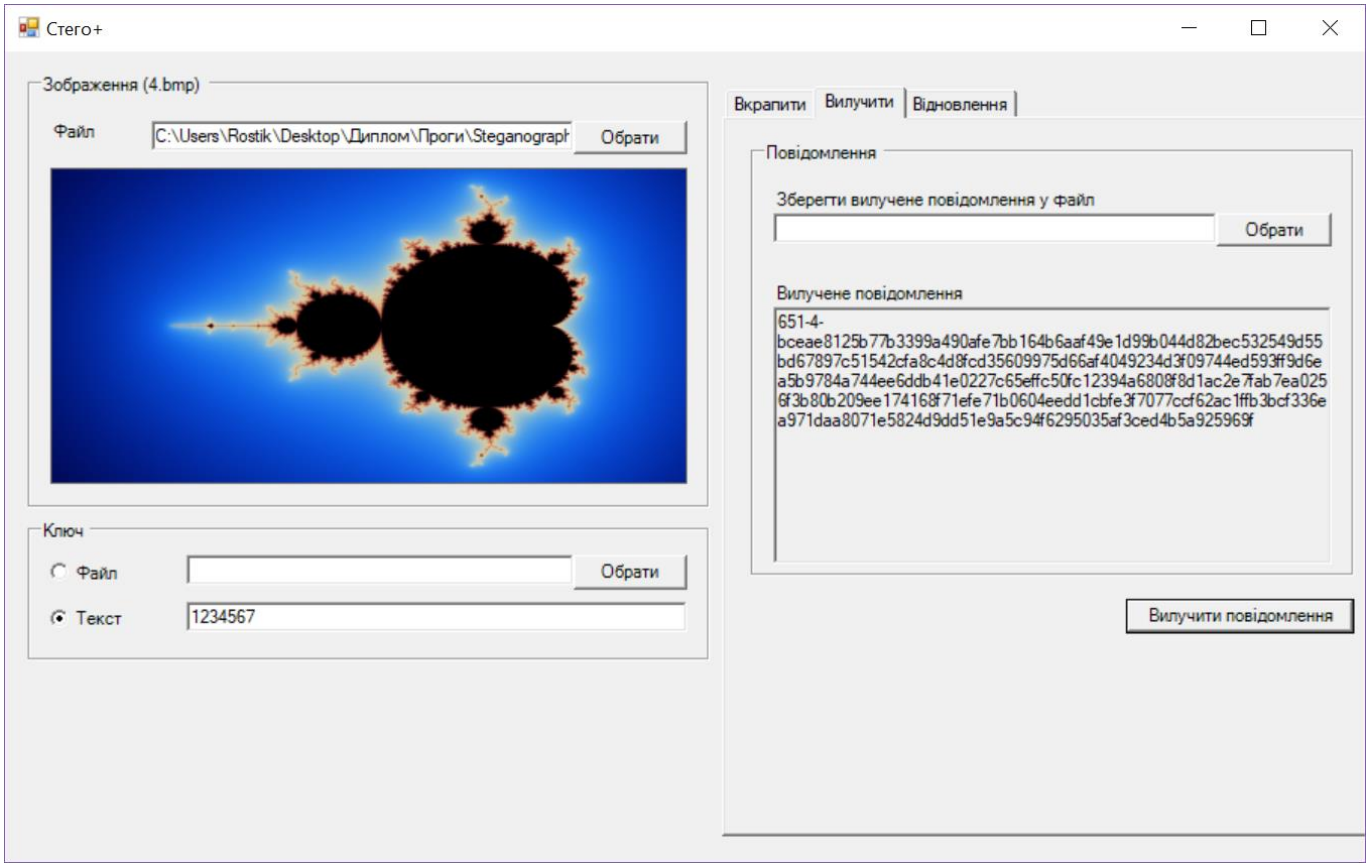

Рисунок 4.10 – Зміст четвертого контейнера після вилучення даних за ключем

На етапі вкраплення було вказано кількість розподілив рівну чотирьом з кількістю розподілів для відновлення – три. Це означає, що після вилучення даних із трьох різних контейнері їх необхідно поєдняти задля того, щоб відновити початкове повідомлення.

Для цього необхідно перейти на закладку «Відновлення» та вставити в текстове поле деяку кількість розподілів, вилучених із контейнерів. Приклад заповнення форми показано на рисунку 4.11.

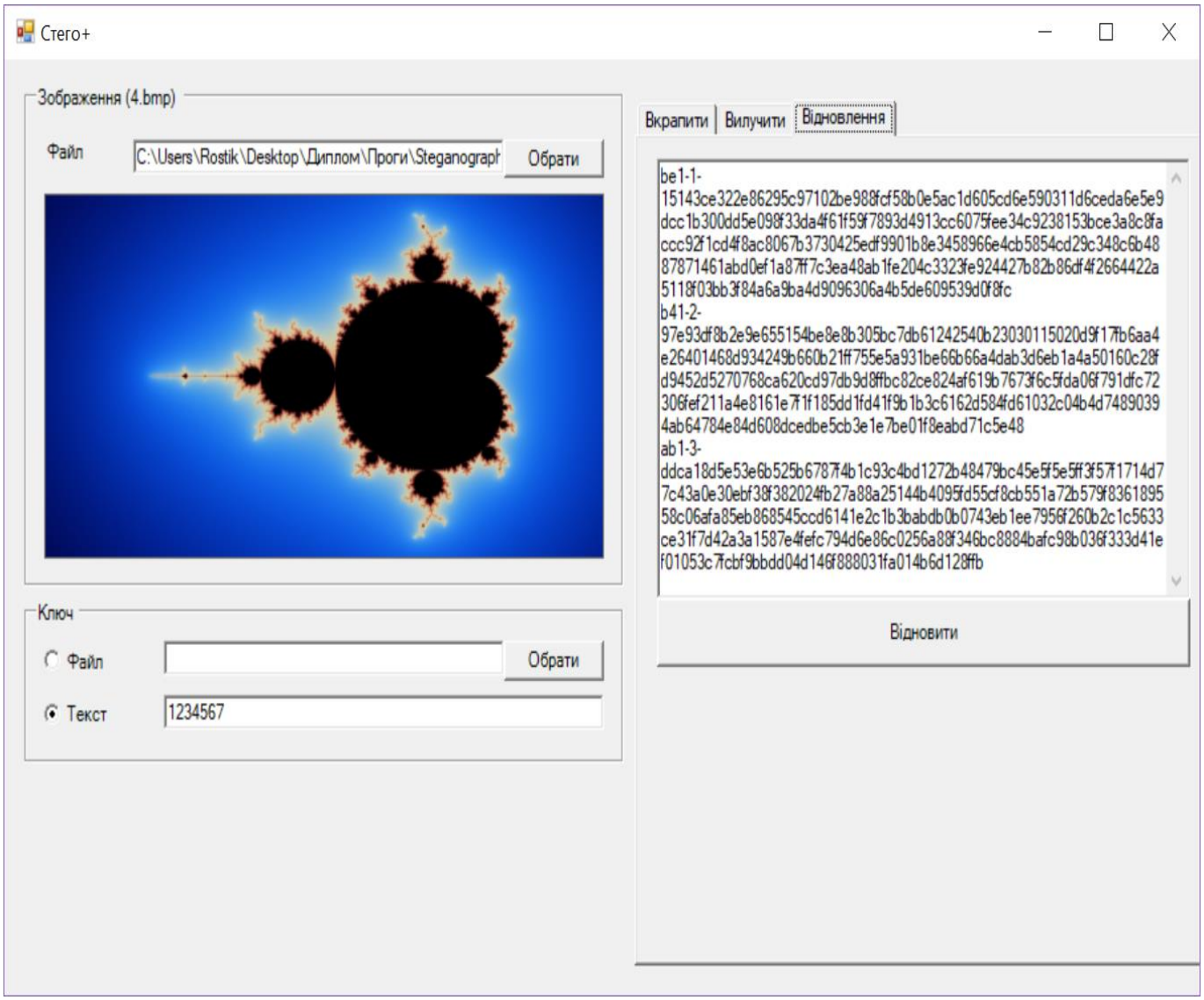

Рисунок 4.11 – Заповнення форми для відновлення повідомлення розподілами

Після перевірки вказаних розподілів, необхідно натиснути кнопку «Відновити». В результаті виконається відновлення початкового повідомлення(секрету) за розподілами, використовуючи інтерполяційний поліном Лагранжа, що використовується у схемі розподілу секрету Шаміра на етапі відновлення. Результат операції продемонстровано на рисунку 4.12.

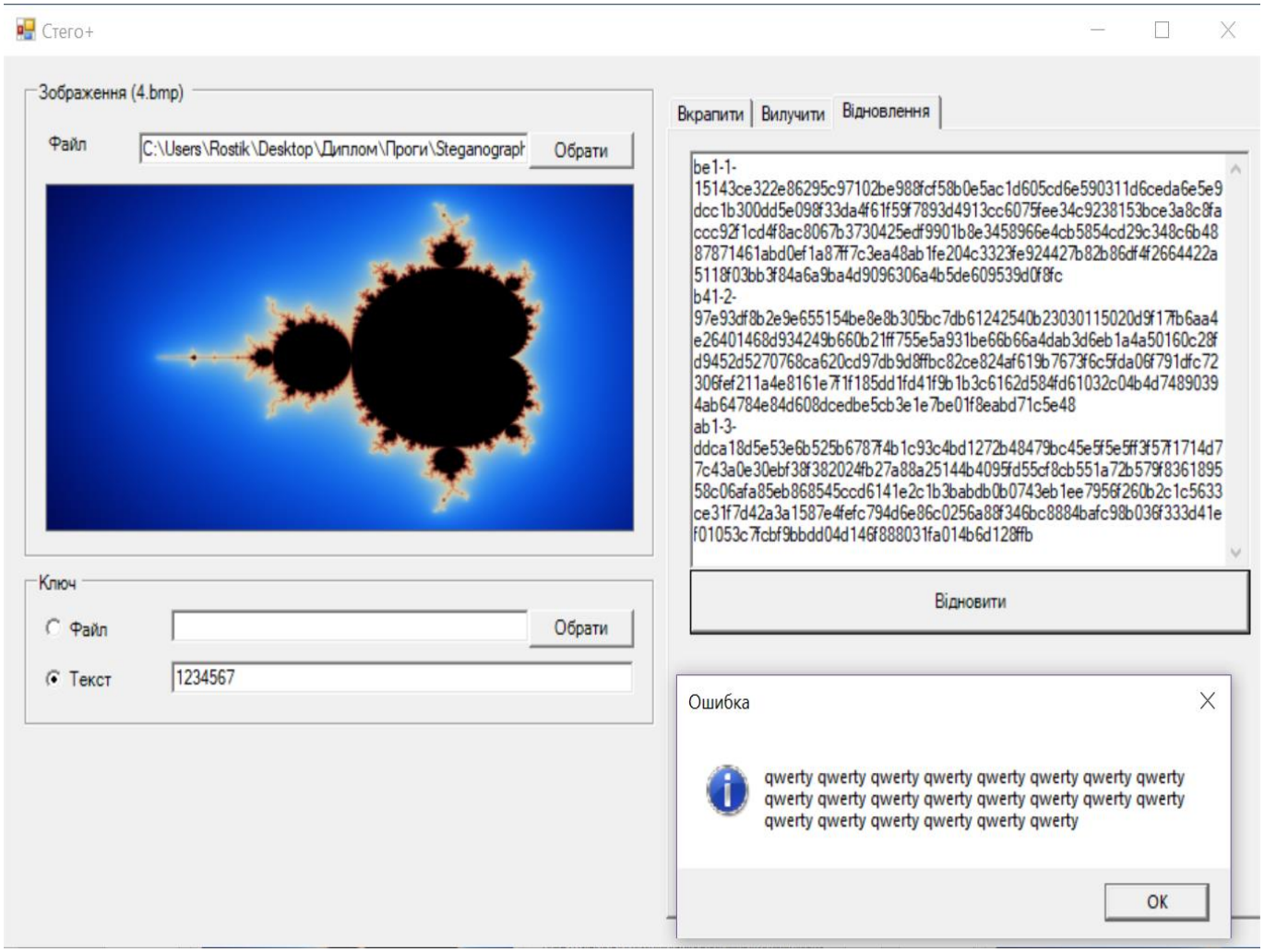

Рисунок 4.12 – Результат відновлення початкового повідомлення трьома секретами

Як можна побачити, відновлене повідомлення повністю співпадає з початковим, що свідчить про правильність роботи застосунку.

Під час тестування збоїв та недоліків у роботі програмного додатку комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення не виявлено, що говорить про високу якість розробки та можливість впровадження у роботу за потребою. Варіанти виключних подій зображено у додатку А (плакат 6).

4.5 Аналіз результатів роботи програми

Для аналізу результатів роботи програми буде здійснено хі-квадрат атаку, а після неї – RS-атаку. Дані атаки є статистичними методами стеганоаналізу, які порівнюють області пікселів, ідеально виявляють наявність повідомлення при звичайному послідовному НЗБ-вкрапленні. Для експерименту візьмемо стеганоконтейнер, в якому вкраплено перший розподіл секрету за схемою Шаміра, а саме:

be1-1-
15143ce322e86295c97102be988fcf58b0e5ac1d605cd6e590311d6ceda6e5e9dcc1b300dd5
e098f33da4f61f59f7893d4913cc6075fee34c9238153bce3a8c8faccc92f1cd4f8ac8067b373
0425edf9901b8e3458966e4cb5854cd29c348c6b4887871461abd0ef1a87ff7c3ea48ab1fe20
4c3323fe924427b82b86df4f2664422a5118f03bb3f84a6a9ba4d9096306a4b5de609539d0f
8fc

Розмір даного сегменту розподілу складає 312 байт, оскільки його довжина – 312 символів, а кодування в системі UTF-8, звідси і отримується розмір у 312 байт. Розмір контейнера складає 2,01 кБ, що становить 2010 байт. Тобто, повідомлення становить 16% від розміру контейнера.

Атака методами хі-квадрат та RS-атака, має нам вказати об’єм інформації, прихованої у контейнері, що подається на обробку методами стеганоаналізу. Результати атак продемонстровано на рисунках 4.13 та 4.14. Для проведення цих двох атак було використано онлайн-ресурс під назвою lsbtools, який є npm-модулем для додатків, що написані на мові JavaScript, а саме: NodeJs-розробки, React, Angular, Vue, VulcanJs. Щоб отримати доступ для проведення атак на контейнери (зображення із вкрапленим повідомленням) необхідно перейти за посиланням https://desudesutalk.github.io/lsbtools.

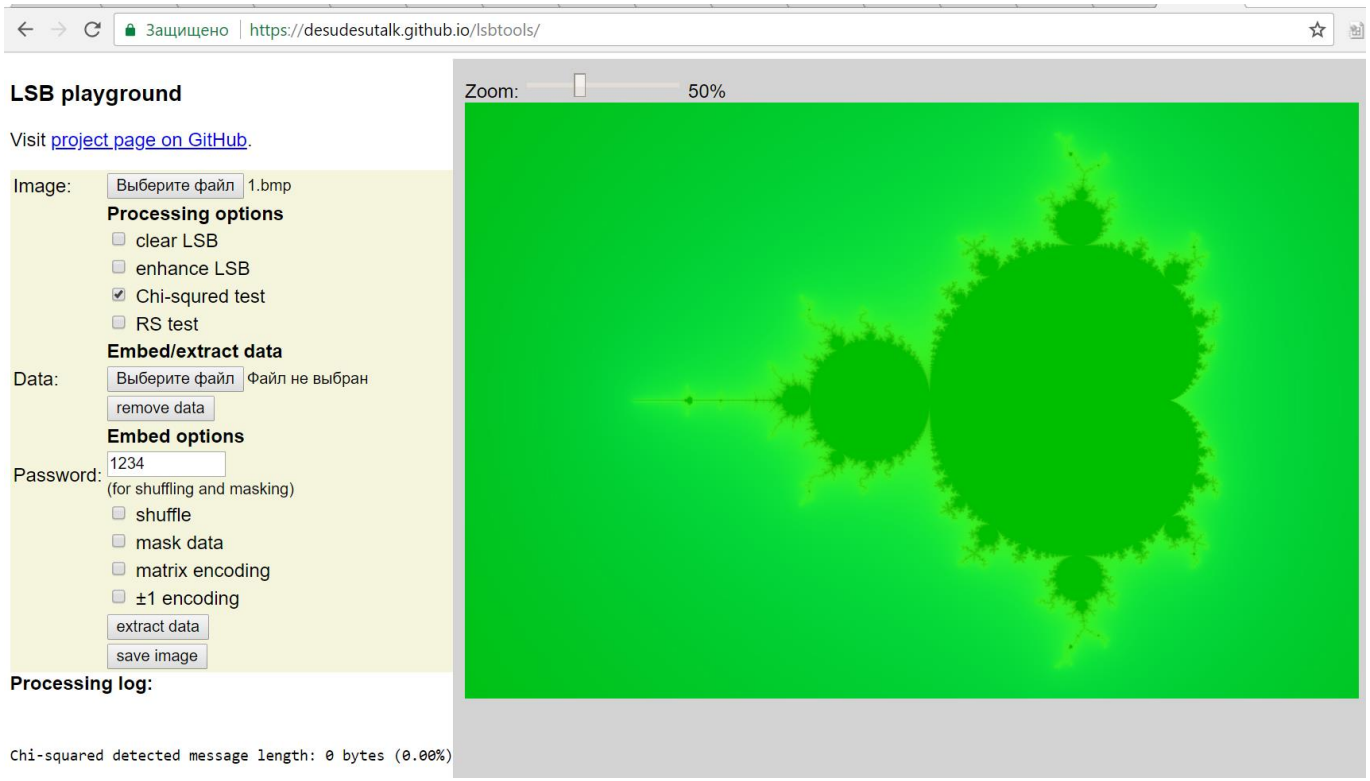

Рисунок 4.13 – Атака стеганоконтейнеру хі-квадрат

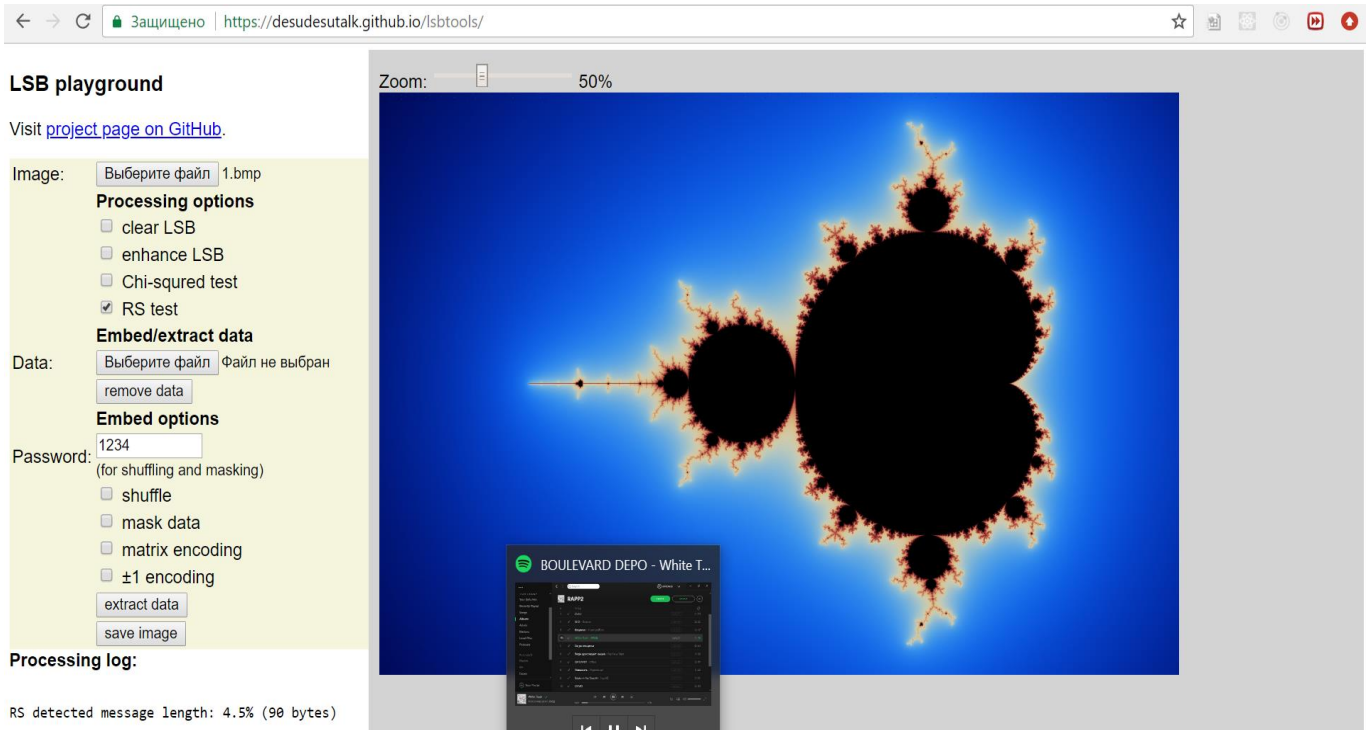

Рисунок 4.14 – RS-атака стеганоконтейнеру

Як бачимо, результати атак хибні, це свідчить про досить надійне приховування для даної довжини повідомлення, яке вкраплено в контейнер. Метод хі-квадрат атаки не виявив вкрапленої інформації, а RS-атака розпізнала лише 4,5% даних від обсягу контейнера, в той час як насправді там знаходилось 16% від розміру контейнеру.

Для порівняння ефективності методу, проробимо ці ж самі кроки, але вкраплення тих самих даних буде здійснюватися звичайним послідовним методо найменш значущого біту. Результати хі-квадрат атаки та RS-атаки на такий контейнер показано на рисунках 4.15 та 4.16 відповідно.

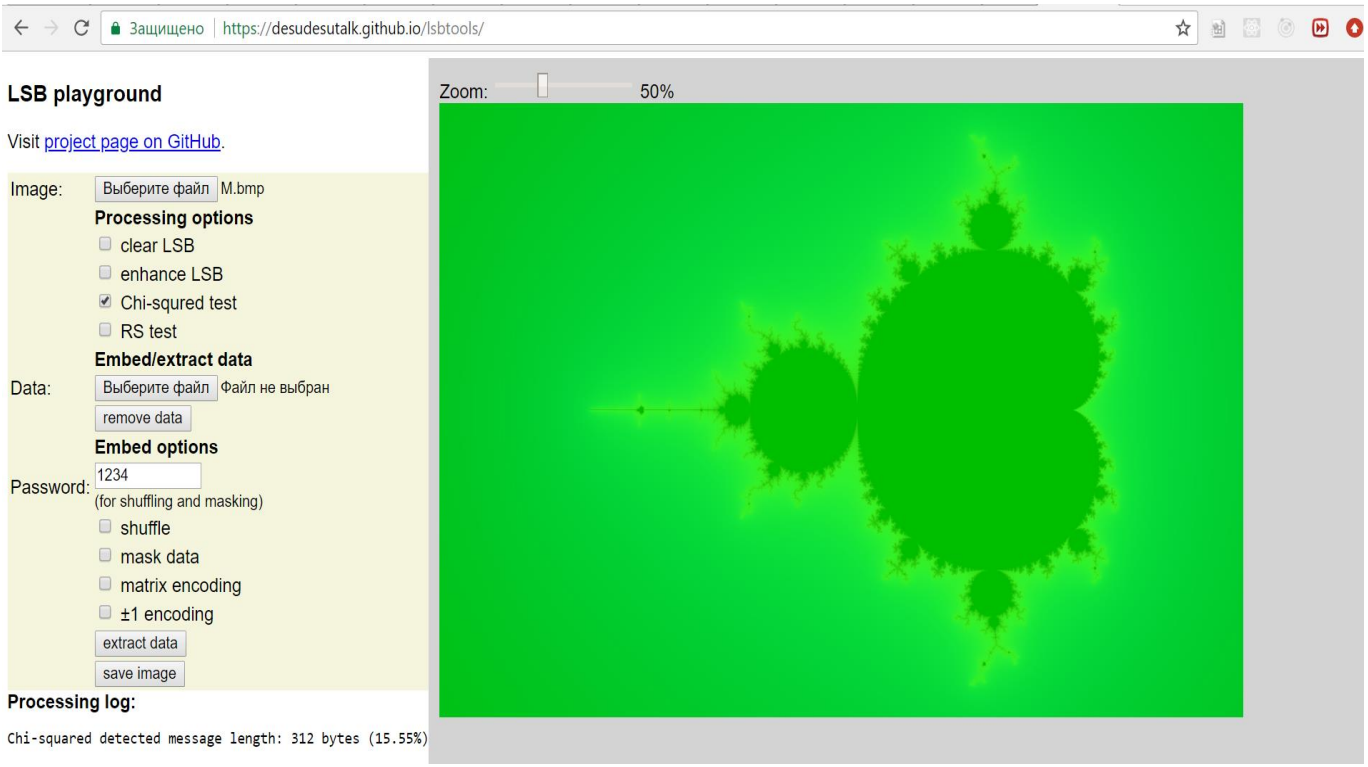

Рисунок 4.15 – Атака стеганоконтейнеру хі-квадрат

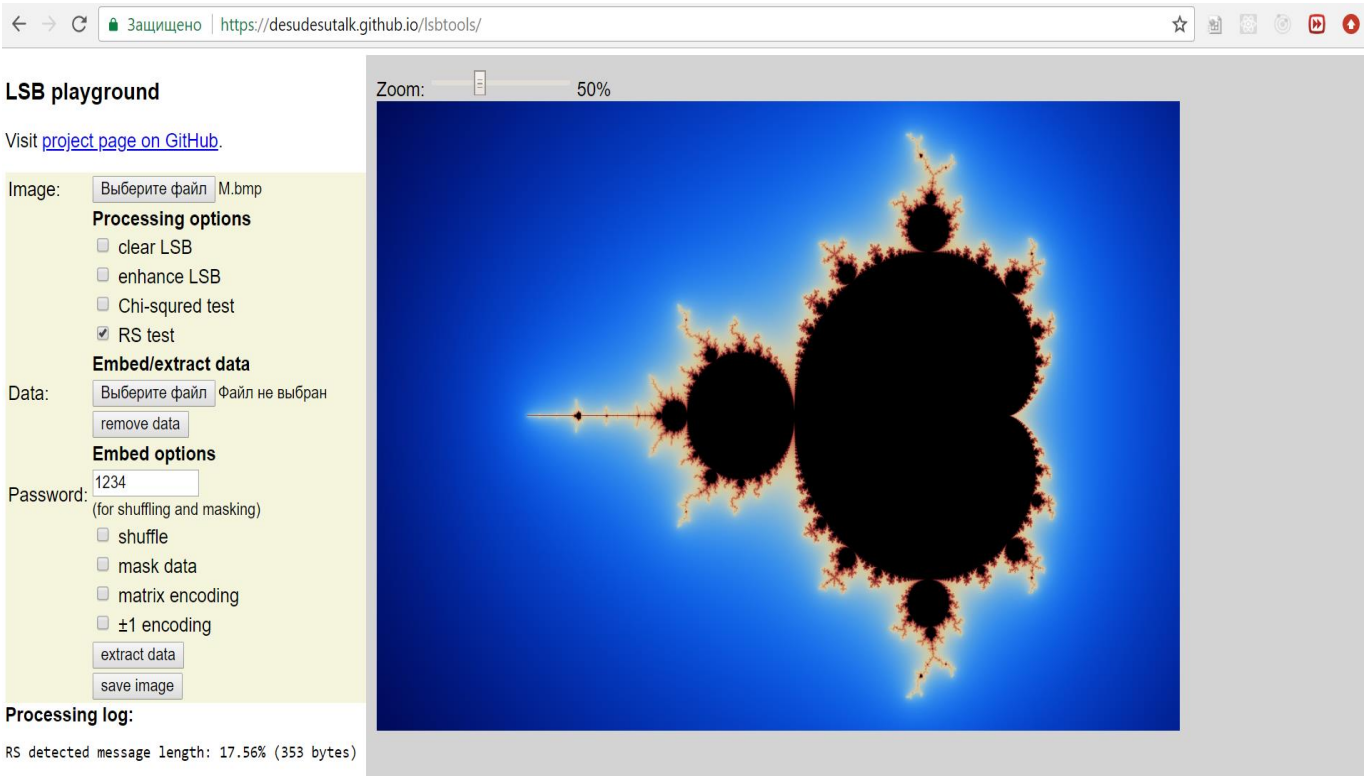

Рисунок 4.16 – RS-атака стеганоконтейнеру

Як бачимо, атаки на контейнер, де відбулося вкраплення послідовним НЗБ, відбулися вдало і виявили наявність повідомлення в контейнері. Хі-квадрат атака точно виявила розмір повідомлення, а RS-атака з невеликою похибкою. Отже, як показує аналіз двох видів атак, запропонований алгоритм краще виконує поставлену задачу.

4.6 Результати досліджень

У межах роботи, було розроблено модифікацію існуючого алгоритму найменш значущого біту для вкраплення інформації в контейнер. Завдяки модифікації, алгоритм коректно працює з повідомленнями, які вкраплюються, підвищуючи стійкість та пропускну здатність. Хоча алгоритм не став ідеальним для передачі даних шляхом вкраплення їх у контейнер, проте не з занадто великими повідомленнями він працює ідеально.

Оскільки вкраплення у випадковому порядку за ключем здійснюється не послідовно з певним кроком, де простежується закономірність, а у непередбачуваному порядку, то це ускладнює завдання аналітикам для виявлення

прихованої інформації в контейнері та її коректного вилучення зловмисниками. В таблицях 4.1 та 4.2 наведено результати RS-атаки та атаки хі-квадрат на стеганоконтейнери, в які було вкраплено інформацію звичайним та модифікованим НЗБ методами в залежності від наповненості контейнера, а саме показано точність атак на контейнери, тобто усереднене відношення виявленого обсягу інформації до справжньої. Для експерименту було взято 500 контейнерів, в які відбувалось вкраплення.

Таблиця 4.1 – Результати хі-квадрат атаки

Наповненість контейнера	НЗБ	Модифікований НЗБ
0	0	0
15	60,74	49,04
30	88,17	52,19
45	91,02	66,38
60	94,31	89,91
75	97,61	93,21
100	100	100

Таблиця 4.2 – Результати RS-атаки

Наповненість контейнера	НЗБ	Модифікований НЗБ
0	0	0
15	47,2	5,37
30	69,39	37,74
45	81,72	56,38
60	95,2	83,48
75	99,14	97,21
100	100	100

Збільшилась значно часова складність алгоритму на відновлення початкового повідомлення, оскільки для відновлення застосовується поліном Лагранжа. Це досягнуто завдяки використанню схеми розподілу секрету Шаміра до початкового

повідомлення, що вкрай суттєво підвищує стійкість контейнеру, оскільки навіть у разі вилучення інформації з контейнеру, її неможливо самотужки відновити до початкового вигляду без інших частин розподілу, що гарантує метод Шаміра. Отже, навіть після вилучення, для зловмисника в контейнері не знайдеться ніякої корисної інформації.

Отже, з рисунків, які зображені у додатку А(Плакат 7) видно, що модифікований алгоритм більш коректно з міркувань пропускної здатності ніж звичайний НЗБ алгоритм, важче точно виявити обсяг вкрапленої інформації атаками та відновлення початкового повідомлення навіть після вилучення інформації у правильному порядку з контейнера. З точки зору апартного забезпечення, цей метод вимагає непомітно більше швидкості, для сучасних комп’ютерів.

Висновки до розділу

У межах розділу здійснено розробку програмного застосунку для вирішення поставленої задачі. Описано основні модулі програми та здійснено тестування засобу. Під час тестування збоїв та недоліків у роботі програмного додатку комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення не виявлено, що говорить про високу якість розробки та можливість впровадження у роботу за потребою.

Обраний метод забезпечує кращу локалізацію особливостей зображень і вимагає менших обчислювальних затрат. Дає можливість здійснювати безпосереднє інтегрування малохвильових коефіцієнтів з відповідними малохвильовими коефіцієнтами оригінального зображення на відповідних частотних рівнях і сприяє істотному підвищенню стійкості зображення із вкрапленням.

ВИСНОВКИ

У даній магістерській дисертації було розроблено модифікований алгоритм НЗБ вкраплення у поєднанні зі схемою розподілу секрету Шаміра. Таке поєднання не випадкове, оскільки модифікованій метод вкраплення підвищує пропускну здатність контейнера, а схема розподілу покращує стійкість, як було показано в ході експериментальних досліджень. Мовою реалізації програмної частини є C#.

В результаті першого етапу розробки було проаналізовано предметну область і розглянуто існуючи методи вирішення проблем, пов'язаних з атаками на стеганоконтейнери та їх основні характеристики, що впливають на передачу вкраплених даних. Було розглянуто принципи побудови контейнерів, принципи вкраплення, відмінності, сильні та слабкі сторони. Також було проведено аналіз можливих способів підвищення стійкості та пропускної здатності цифрових контейнерів.На основі проведених дослідження було прийнято рішення використовувати поєднання двох алгоритмів, один з яких модифікований, задля покращення характеристик вихідного стеганоконтейнера.

Оскільки вкраплення у випадковому порядку за ключем здійснюється не послідовно з певним кроком, де простежується закономірність, а у непередбачуваному порядку, то це ускладнює завдання аналітикам для виявлення прихованої інформації в контейнері та її коректного вилучення зловмисниками. Було наведено результати RS-атаки та атаки хі-квадрат на стеганоконтейнери, в які було вкраплено інформацію звичайним та модифікованим НЗБ методами в залежності від наповненості контейнера, а саме показано точність атак на контейнери, тобто усереднене відношення виявленого обсягу інформації до справжньої. Для експерименту було взято 500 контейнерів, в які відбувалось вкраплення.

З огляду на складність спроектованої системи було використано об'єктно-орієнтоване програмування. Даний підхід дозволив спростити процес розширення логіки роботи програмного забезпечення, зробив можливою вчасну обробку усіх вкраплених або вилучених даних.В рамках опису розробленої архітектури побудованого комплексу наведено схему типової стеганосистеми, діаграму

компонентів, діаграми класів, на діаграмі послідовності продемонстровано ітерацію роботи застосунку, також надано детальний опис розробленого алгоритму у вигляді блок-схеми. Також наведена інструкція з використання застосунку та уникнення можливих помилок, які можуть бути викликані через неуважність користувача. Також наведено додаткові можливості використання розробленого комплексу на прикладі передачі текстової інформації за допомогою методів стеганографії та криптографії. Наведено приклади атак на вихідні контейнери застосунку.

З огляду на вище наведене можна зробити висновок, що в результаті проведеної роботи було досягнуто поставлену мету.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мельник С.В. Світові тенденції розвитку цифрової стеганографії в контексті завдань за-безпечення інформаційної безпеки держави / С.В.Мельник, С.В.Кондакова // Актуальні проблеми управління інформаційною безпекою держави : зб. матер. наук.-практ. конф. – К. : Наук.-вид. відділ НА СБ України, 2010. – С. 134-138.
2. Конахович Г.Ф. Компьютерная стегано-графия. Теория и практика / Г.Ф.Конахович, А.Ю.Пузыренко. – К. : МК-Пресс, 2006. – 288 с.
3. Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Туринцев. – М. : СОЛОН-Пресс, 2002. – 272 с.
4. Быков С.Ф., Мотуз О.В. Основы стегоанализа.// Защита информации. Конфидент. – СПб.: 2000, № 3. – С. 38-41.
5. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – М.: Солон-Пресс, 2002. – 272 с.
6. Елтышева Е.Ю., Фионов А.Н. Построение стегосистемы на базе растровых изображений с учетом статистики младших бит // Вестник СибГУТИ. – 2009. № 1. – С. 67-84.
7. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. – К.: МК-Пресс, 2006. – 288 с.
8. Жилкин М. Ю. Стегоанализ графических данных на основе методов сжатия // Вестник СибГУТИ. – 2008. № 2. – С. 62–66.
9. Кувшинов С.С. Методы и алгоритмы сокрытия больших объемов данных на основе стеганографии / Диссертация на соискание ученой степени кандидата технических наук. – Санкт-Петербург. 2010. – 116 с.
10. Бернет С., Пейн С.: Криптография. Официальное руководство RSA Security – М. «Бином», 2012. – 325 с.
11. Венбо Мао Современная криптография: теория и практика = Modern Cryptography: Theory and Practice. – М.: «Вильямс», 2005. – С. 768.

12. Воробьев В.И., Грибунин В.Г. Теория и практика вейвлет-преобразования. – СПб: ВУС, 2009. – 325 с.
13. Зима В.: Безопасность глобальных сетевых технологий – «БХВ-Петербург», 2011. – 344 с.
14. Нильс Фергюсон, Брюс Шнайер Практическая криптография : Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – М.: «Диалектика», 2012. – С. 432.
15. Павлов К.А Компьютерная безопасность. Криптографические методы защиты. ДМК Москва, 2010. – 233 с.
16. Ростовцев А.Г. , Михайлова Н.В. Методи криптоаналізу класичних шифрів. – К.: «Наука», 2012. – С. 142.
17. Саломан А. Криптографія з відкритим ключем. – К.: «Наука», 2013. – 342 с.
18. Серов Р.Е., Гончаров В.В., Основы современной криптографии – Москва, Горячая линия – Телеком, 2011. – 443 с.
19. Столлингс В. Криптография и защита сетей: теория и практика. М: Вильямс. 2001. Пер. с англ. – 235 с.
20. Чмора А.Л. Сучасна прикладна криптографія. 2-е вид., Стер. – М.: Геліос АРВ, 2012. – 256 с.
21. Шнайер, Брюс. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си – М.: Издательство ТРИУМФ, 2002 – 816 с.
22. Mallat S. A Theory For Multiresolution Signal Decomposition: The Wavelet Representation / / IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. – Vol. 11. – P. 674-693.
23. Shapiro J. Embedded Image Coding Using Zerotrees Of Wavelet Coefficients / / IEEE Transactions on Signal Processing, 1993. – Vol. 41, No. 12.
24. Said A., Pearlman W. A New Fast And Efficient Image Codec Based On Set Partitioning in Hierarchical Trees / / IEEE Transactions on Circuits and Systems for Video Technology, 1996. – Vol. 6. – P. 243-250.

25. Antonini M., Barlaud M., Mathieu P., Daubechies I. Image Coding Using Wavelet transform / / IEEE Transactions On Image Processing, 1992. – Vol. 1, № 2. – P. 205-220
26. Аграновский А.В. Основы компьютерной стеганографии / А.В. Аграновский, П.Н. Девянин, Р.А. Хади, А.В. Черемушкин. – М: Радио и связь, 2003. – 152 с.
27. Кошкина Н.В. Стеганоаналіз цифрових зображень із застосуванням контрольного вкраплення // Матеріали з Міжнар. наук.-техн. конф. «Захист інформації і безпека інформаційних систем», 5–6 черв. 2014. – Львів: Львівська політехніка, 2014. – С. 98– 100.
28. Стеганоанализ изображений в формате jpeg на базе атаки контрольным внедрением / Н.В. Кошкина // Управляющие системы и машины. — 2014. — № 4. — С. 3-9, 17.
29. Кошкина Н.В. Стеганоанализ бесключевых стеганосистем на основе атаки контрольным внедрением / Н.В. Кошкина // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2014. – № 6. – С. 137–144.
30. Ахмад Х.М. Введение в цифровую обработку речевых сигналов / Х.М. Ахмад, В.Ф. Жирков. – Владимир: Издво Владим. гос. ун-та, 2007. – 192 с.
31. Кошкина Н.В. Обзор и классификация методов стеганоанализа / Н.В. Кошкина // УСИМ. – 2015. – № 3. – С. 3–12.
32. Кошкіна Н.В. Методи стеганоаналізу з навчанням та класифікацією за характеристичними векторами / Н.В. Кошкіна // Праці міжнар. конф. “Питання оптимізації обчислень-XL”. – Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2015. – С. 153–154.
33. Капуста А.М. Методы статистической классификации в задаче обнаружения встраивания информации / А.М. Капуста // Сб. работ 68-й науч. конф. студентов и аспирантов Белорусского гос. ун-та 16-19 мая 2011 г.: в 3-х ч.: ч. 1. – Минск, 2011. – С. 117–120
34. Защелкин К.В. Решение проблемы классификации блоков контейнера при jpeg-атаке на стеганографический метод Бенгама-Мемона-Эо-Юнг / К.В.

Защелкин, А.А. Ищенко, Е.Н. Иванова // Радіоелектронні і комп’ютерні системи. – 2014. – № 6 (70). – С. 164–168.

35. Zadiraka V. Spectral methods of computer steganography problem decision / V. Zadiraka, N. Koshkina // Methods of effective protection of information flows /ed. by V. Zadiraka, Y. Nykolaichuk. – Ternopil: Ternograf, 2014. – P. 96–120.
36. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – К.: МК-Пресс, 2006. – 288 с.
37. Кошкіна Н.В. Інформаційно-теоретична модель безпеки стеганографічних систем / Н.В. Кошкіна // Поступ в науку. – 2011. – №6, Т.1. – С.117–120.
38. Хорошко В.А. Введение в компьютерную стеганографию / В.А. Хорошко, М.Е. Шелест. – Киев: Національний Авіаційний Університет, 2002. – 152 с.
39. Задирака В.К. К вопросу стойкости стеганосистемы при пассивных атаках / В.К. Задирака, Л.Л. Никитенко // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2009. – № 2. – С. 132 – 138.
40. Кошкіна Н.В. Ефективні спектральні алгоритми для вирішення задач цифрової стеганографії: дис. … канд. фіз.-мат. наук: 01.05.01 / Н.В. Кошкіна. – Київ, 2005.– 139 с
41. Клопов В.А. Основы компьютерной стеганографии / В.А. Клопов, О.В. Мотуз // Конфидент. – 1997. – №4. – С.43–48.
42. Кустов В.Н. Методы встраивания скрытых сообщений / В.Н. Кустов, А.А. Федчук // Конфидент. – 2000. – № 3. – С.34–37.
43. Задірака В.К. Спектральні алгоритми комп’ютерної стеганографії / В.К. Задірака, С.С. Мельнікова, Н.В. Бородавка //Искусственный интеллект. – 2002. – № 3. – С. 532 –541.
44. Швидченко И.В. Методы стеганоанализа для графических файлов / И.В. Швидченко // Искусственный интеллект. – 2010. – № 4. – С. 697–705.
45. Швідченко І.В. Аналіз програмного забезпечення зі стеганоаналізу / І.В. Швідченко //Искусственный интеллект. – 2012. – №3. – С. 487–495.

46. Li F. JPEG steganalysis with high-dimensional features and bayesian ensemble classifier / F. Li, X. Zhang, B. Chen, G. Feng //IEEE signal processing letters. – 2013. – Vol. 20, № 3. – P. 233–236.
47. Яне Б. Цифровая обработка изображений / Б. Яне. – М.: Техносфера, 2007. – 583 с
48. Вовк О.О. Сравнительный анализ устойчивости к атакам стеганографических методов скрытия информации / О.О. Вовк, А.А. Астраханцев // Мат. 9-й Межд. мол. научно-техн. конф. «Современные проблемы радиотехники и телекоммуникаций РТ-2013». – 2013. – с. 153.
49. Кошкина Н.В. О методе защиты интеллектуальной собственности на основе выделения точечных особенностей изображения / Н.В. Кошкина //Захист інформації. – 2007. – №4. – С. 52–63.
50. Avcibas I. Image steganalysis with binary similarity measures / I. Avcibas¸ M. Kharrazi, N.D. Memon, B. Sankur //EURASIP Journal on Applied Signal Processing. – 2005. – P. 2749–2757.
51. Кошкіна Н.В. Стійкі до активних атак методи комп'ютерної стеганографії / Н.В. Кошкіна // Вісн. НАН України. – 2013. – № 4. – С. 61–66.
52. Задирака В.К. Статистический анализ систем с цифровыми водяными знаками / В.К. Задирака, Н.В. Кошкина, Л.Л. Никитенко // Искусственный интеллект.– 2008.– № 3. – С. 315–324.
53. Мелешко Е.В. Метод встраивания двухуровневых цифровых водяных знаков в медиафайлы для защиты авторских прав / Е.В. Мелешко // Збірник наукових праць Харківського університету Повітряних Сил. – 2013. – № 4. – С. 127-131.
54. Suresh A. Image Texture Classification using Gray Level Co-Occurrence Matrix Based Statistical Features / A. Suresh, K.L. Shunmuganathan // European Journal of Scientific Research. – 2012. – Vol.75, № 4. – P. 591–597
55. Voloshynovskiy S.V. Visual communications with side information via distributed printing channels: extended multimedia and security perspectives / S.V. Voloshynovskiy, O. Koval, F. Deguillaume, T. Pun // Proc. of SPIE: Security,

Steganography, and Watermarking of Multimedia Contents VI, San Jose, USA, January 2004. – P. 428–445.

56. Lin C.-Y. Distortion Modeling and Invariant Extraction for Digital Image Printand-Scan Process [Електронний ресурс] / C.-Y. Lin, S.-F. Chang // Intl. Symp. on Multimedia Information Processing, Taipei, December 1999. – Режим доступу: http://www.ee.columbia.edu/ln/dvmm/publications/99/cylin-modelscan.pdf.
57. Терещенко А.Н. Реализация операции умножения с использованием преобразования Уолша / А.Н. Терещенко, С.С. Мельникова, Л.А. Гнатив, В.К. Задирака, Н.В. Кошкина // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2010. – №2. – С. 102–126.
58. Задирака В.К. К вопросу стойкости стеганосистемы при пассивных атаках / В.К. Задирака, Л.Л. Никитенко // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2009. – № 2. – С. 138 – 139.
59. Гнатив Л.А. Методы синтеза эффективных ортогональных преобразований высокой и низкой корреляции и их быстрых алгоритмов для кодирования и сжатия цифровых изображений / Л.А. Гнатив, Е.С. Шевчук //Кибернетика и системный анализ. – 2002. № 6. С. 104–117.
60. Yang S. Quantization-Based Digital Audio Watermar-king in Discrete Fourier Transform Domain / S. Yang, W. Tan, Y. Chen, W. Ma // Journal of Multimedia. – 2010. – Vol. 5, № 2. – P. 151–158.
61. Поліновський В.В. Інформаційна технологія для досліджень методів стеганографії і стеганоаналізу / В.В. Поліновський, В.Ю. Корольов, В.А. Герасименко, М.Л. Горинштейн // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2011. – №5. – С. 236–242.
62. Manjula Devi T.H. Detecting original image using histogram, DFT and SVM / T.H. Manjula Devi, H.S. Manjunatha Reddy, K.B. Raja, K.R. Venugopal, L.M. Patnaik //Intern. journal of recent trends in engineering. – 2009. – Vol. 1, №1. – P. 367–371.
63. Sheikhan M. Blind image steganalysis via joint co-occurrence matrix and statistical moments of contourlet transform / M. Sheikhan, M.S. Moin, M. Pezhmanpour //10th Int. Conf. on Intelligent Systems Design and Applications. – 2010. – P. 368–372.

64. Yang X. Universal image steganalysis based on wavelet packet decomposition and empirical transition matrix in wavelet domain / X. Yang, Y. Lei, X. Pan, J. Liu // International forum on computer science-technology and applications. – 2009. – Vol. 2. – P. 179–182.

65. Кошкина Н.В. Защита космических и астрономических изображений методами компьютерной стеганографии / Н.В. Кошкина, О.Ю. Никитина // Праці IV міжнар. наук.-техн. конф. “Гіротехнології, навігація, керування рухом і конструю-вання авіаційно-космічної техніки”, Ч. 2. – Київ: НТУУ «КПІ». – 2007. – С. 397–402.

66. Кошкіна Н.В. До питання часо-частотного аналізу сигналів в задачах комп’ютерної стеганографії / Н.В. Кошкіна // Праці міжнар. конф. “Питання оптимізації обчислень-XXXVI. Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2011. – Том 1. – С. 351–355.

67. Barni M. A DWT-based technique for spatio-frequency masking of digital signatures / M. Barni, F. Bartolini, V. Cappellini, A. Lippi, A. Piva // Proc. of the 11th SPIE Annual Symposium, Electronic Imaging, Security and Watermarking of Multimedia Contents. – 1999. – Vol. 3657. – P. 31-39.

68. Защелкин К.В. Решение проблемы классификации блоков контейнера при jpeg-атаке на стеганографический метод Бенгама-Мемона-Эо-Юнг / К.В. Защелкин, А.А. Ищенко, Е.Н. Иванова // Радіоелектронні і комп’ютерні системи. – 2014. – № 6 (70). – С. 164–168.

69. Вовк О.О. Сравнительный анализ устойчивости к атакам стеганографических методов скрытия информации / О.О. Вовк, А.А. Астраханцев // Мат. 9-й Межд. мол. научно-техн. конф. «Современные проблемы радиотехники и телекоммуникаций РТ-2013». – 2013. – с. 153.

70. Simitopoulos D. Robust Image Watermarking Based on Generalized Radon Transformations / D. Simitopoulos, D.E. Koutsonanos, M.G. Strintzis // Circuits and Systems for Video Technology. – 2003. – Vol. 13, №8. – P. 732–745.

71. Chiu Y.-C. Copyright Protection against Print-and-Scan Operations by Watermarking for Color Images Using Coding and Synchronization of Peak Locations in Frequency

Domain / Y.-C. Chiu, W.-H. Tsai // Journal of Information Science and Engineering. – 2006. – Vol. 22, № 3. – P. 483–496.

72. Lowe D.G. Distinctive image features from scale-invariant keypoints / D.G. Lowe // International journal of computer vision. – 2004. – Vol. 60, №2. – P. 91–110.
73. Романчук Р.О., Поліщук А.О. Приховування інформації використовуючи аудіо стеганографію / Матеріали міжнародної наукової конференції «Актуальні наукові дослідження в сучасному світі», 27-28 лютого 2018 р. – С. 36-42.
74. Романчук Р.О., Поліщук А.О. Вплив стеганографії та схеми розподілу секрету зображень на безпеку криптографічного ключа / Матеріали міжнародної наукової конференції «Актуальні наукові дослідження в сучасному світі», 26-27 грудня 2017 р. – С. 27-33.

ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ

ПЛАКАТ 1 Схема типової стеганосистеми

ПЛАКАТ 2 Блок-схема роботи алгоритму

ПЛАКАТ 3 Діаграма послідовності

ПЛАКАТ 4 Діаграма діяльності

ПЛАКАТ 5 Діаграма класів

ПЛАКАТ 6 Екранні форми

ПЛАКАТ 7 Результати дослідження